  
  
**THỰC TẬP CHUYÊN ĐỀ VÀ ĐỒ ÁN CHUYÊN MÔN**

***ĐỀ TÀI***

**TÌM HIỂU BÀI TOÁN QUẢN LÝ**

**HỆ THỐNG PHÒNG TRỌ**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | **Th.S Trịnh Văn Chung** |
| **Sinh viên thực hiện:** | **Nguyễn Văn Thịnh** |
| **Mã sinh viên:** | **2310900101** |
| **Lớp:** | **K32CNT1** |
| **Khóa:** | **2023-2027** |

**@FITNTU-PJ3** 12/2025

# MỤC LỤC

[MỤC LỤC 2](#_Toc217914289)

[LỜI CẢM ƠN 5](#_Toc217914290)

[DANH MỤC BẢNG BIỂU 6](#_Toc217914291)

[DANH MỤC HÌNH ẢNH 7](#_Toc217914292)

[CHƯƠNG 1: NGHIÊN CỨU CƠ SỞ LÝ THUYẾT 8](#_Toc217914293)

[1.1. Tổng quan về lập trình thiết kế giao diện website 8](#_Toc217914294)

[1.1.1 Giới thiệu HTML5 8](#_Toc217914295)

[1.1.2 Giới thiệu về CSS3 8](#_Toc217914296)

[1.1.3 Tổng quan về ngôn ngữ lập trình Javascript 10](#_Toc217914297)

[1.1.4 Thymeleaf Template Engine 11](#_Toc217914298)

[1.2. Tổng quan về hệ quản trị cơ sở đữ liệu MySQL 11](#_Toc217914299)

[1.2.1 Giới thiệu về MySQL 12](#_Toc217914300)

[1.2.2 Ưu điểm 12](#_Toc217914301)

[1.2.3 Nhược điểm 13](#_Toc217914302)

[1.3. Tổng quan về ngôn ngữ lập trình phía máy chủ 13](#_Toc217914303)

[1.3.1 Giới thiệu ngôn ngữ lập trình Java 13](#_Toc217914304)

[1.3.2 Giới thiệu Spring Boot Framework (Spring MVC, Spring Data JPA, Spring Security) 14](#_Toc217914305)

[CHƯƠNG 2: TỔNG QUAN VỀ ĐỀ TÀI 15](#_Toc217914306)

[2.1. Đặt vấn đề 15](#_Toc217914307)

[2.2. Hệ thống hiện tại 15](#_Toc217914308)

[2.3. Hệ thống đề xuất 15](#_Toc217914309)

[2.4. Giới hạn của hệ thống 16](#_Toc217914310)

[2.5. Yêu cầu về phần cứng và phần mềm 17](#_Toc217914311)

[2.5.1 Yêu cầu tối thiểu 17](#_Toc217914312)

[2.5.2 Yêu cầu đề nghị 17](#_Toc217914313)

[CHƯƠNG 3: 18](#_Toc217914314)

[PHÂN TÍCH YÊU CẦU KHÁCH HÀNG VÀ THIẾT KẾ HỆ THỐNG 18](#_Toc217914315)

[3.1. Người dùng của hệ thống 18](#_Toc217914316)

[3.2. Chức năng của hệ thống 18](#_Toc217914317)

[3.2.1 Guest functions (Chức năng dành cho khách vãng lai) 18](#_Toc217914318)

[3.2.2 Member functions (Chức năng dành cho thành viên) 19](#_Toc217914319)

[3.2.3 Admin functions (Chức năng dành cho quản trị viên) 21](#_Toc217914320)

[3.3. Systeme Designs (Thiết kế hệ thống) 24](#_Toc217914321)

[3.3.1 Thực thể users (người dùng) 24](#_Toc217914322)

[3.3.2 Thực thể tenants (Người thuê phòng) 24](#_Toc217914323)

[3.3.3 Thực thể hostels (Khu nhà trọ) 24](#_Toc217914324)

[3.3.4 Thực thể rooms (Phòng trọ) 25](#_Toc217914325)

[3.3.5 Thực thể amenities (Tiện nghi) 25](#_Toc217914326)

[3.3.6 Thực thể room\_amenities (Tiện nghi của phòng) 25](#_Toc217914327)

[3.3.7 Thực thể contracts (Hợp đồng thuê) 26](#_Toc217914328)

[3.3.8 Thực thể contract\_members (Thành viên hợp đồng) 26](#_Toc217914329)

[3.3.9 Thực thể services (Dịch vụ) 26](#_Toc217914330)

[3.3.10 Thực thể meter\_readings (Chỉ số điện) 27](#_Toc217914331)

[3.3.11 Thực thể bills (Hóa đơn) 27](#_Toc217914332)

[3.3.12 Thực thể bill\_items (Chi tiết hóa đơn) 27](#_Toc217914333)

[3.3.13 Thực thể payments (Thanh toán) 27](#_Toc217914334)

[3.3.14 Thực thể booking\_requests (Yêu cầu xem phòng) 28](#_Toc217914335)

[3.3.15 Thực thể feedbacks (Phản ánh / bảo trì) 28](#_Toc217914336)

[3.4. Database Design (Thiết kế cơ sở dữ liệu) 29](#_Toc217914337)

[3.4.1 Bảng users (người dùng) 29](#_Toc217914338)

[3.4.2 Bảng tenants (Người thuê phòng) 29](#_Toc217914339)

[3.4.3 Bảng hostels (Khu nhà trọ) 30](#_Toc217914340)

[3.4.4 Bảng rooms (Phòng trọ) 30](#_Toc217914341)

[3.4.5 Bảng amenities (Tiện nghi) 31](#_Toc217914342)

[3.4.6 Bảng room\_amenities (Tiện nghi của phòng) 31](#_Toc217914343)

[3.4.7 Bảng contracts (Hợp đồng thuê) 31](#_Toc217914344)

[3.4.8 Bảng contract\_members (Thành viên hợp đồng) 32](#_Toc217914345)

[3.4.9 Bảng services (Dịch vụ) 32](#_Toc217914346)

[3.4.10 Bảng meter\_readings (Chỉ số điện nước) 32](#_Toc217914347)

[3.4.11 Bảng bills (Hóa đơn) 33](#_Toc217914348)

[3.4.12 Bảng bill\_items (Chi tiết hóa đơn) 33](#_Toc217914349)

[3.4.13 Bảng payments (Thanh toán) 34](#_Toc217914350)

[3.4.14 Bảng booking\_requests (Yêu cầu xem phòng) 34](#_Toc217914351)

[3.4.15 Bảng feedbacks (Phản ánh / bảo trì) 34](#_Toc217914352)

[3.5 Table RelationShip Diagram (Mô hình quan hệ bảng) 35](#_Toc217914353)

[3.6 SiteMap (Sơ đồ cấu trúc site) 36](#_Toc217914354)

[3.6.1 Site map (Cấu trúc trang): Dành cho khách vãng lai 36](#_Toc217914355)

[3.6.2 Sitemap (Cấu trúc trang): Dành cho login admin 36](#_Toc217914356)

[3.6.3 Dành cho quản trị viên 37](#_Toc217914357)

[3.7 Algorithms (Giải thuật) 38](#_Toc217914358)

[3.7.1 Đăng ký 38](#_Toc217914359)

[3.7.2 Đăng nhập 39](#_Toc217914360)

[3.7.3 Đăng xuất 39](#_Toc217914361)

[3.7.4 Tạo hóa đơn 40](#_Toc217914362)

[3.7.5 User gửi yêu cầu thuê đến Admin 41](#_Toc217914363)

[3.7.6 User gửi yêu cầu thanh toán và chờ thông báo gmail 42](#_Toc217914364)

[3.7.7 Admin duyệt thanh toán và gửi thông báo gmail 43](#_Toc217914365)

[CHƯƠNG 4: XÂY DỰNG HỆ THỐNG 44](#_Toc217914366)

[4.1. Xây dựng phần mềm 44](#_Toc217914367)

[Mã nguồn chương trình 44](#_Toc217914368)

[4.2 Một số giao diện người dùng và chức năng cụ thể 44](#_Toc217914369)

[4.2.1 Giao diện trang chủ 44](#_Toc217914370)

[4.2.2 Giao diện tra cứu hoá đơn theo phòng 45](#_Toc217914371)

[4.2.3 Giao diện thanh toán hoá đơn 45](#_Toc217914372)

[4.2.4 Giao diện gửi phản ánh 46](#_Toc217914373)

[4.2.5 Giao diện đăng nhập 46](#_Toc217914374)

[4.2.6 Giao diện quản trị viên 47](#_Toc217914375)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 53](#_Toc217914376)

[5.1. Kết luận 53](#_Toc217914377)

[5.2 Checklist (bảng kiểm tra chức năng) 54](#_Toc217914378)

[5.2.1 Kiểm tra dữ liệu nhập 54](#_Toc217914379)

[5.2.2 Kiểm tra các liên kết 54](#_Toc217914380)

[DANH MỤC TÀI LIỆU THAM KHẢO 56](#_Toc217914381)

# LỜI CẢM ƠN

Trước hết, em xin gửi lời cảm ơn chân thành đến Ban Giám hiệu trường Đại học Nguyễn Trãi, cùng toàn thể các thầy cô trong khoa đã tạo điều kiện thuận lợi để chúng em được học tập, nghiên cứu và thực hiện đồ án này.

Đặc biệt, chúng em xin bày tỏ lòng biết ơn sâu sắc tới Thầy Trịnh Văn Chung – giảng viên hướng dẫn, người đã tận tình chỉ bảo, định hướng và đóng góp nhiều ý kiến quý báu giúp chúng em hoàn thành tốt đồ án.

Chúng em cũng xin cảm ơn các bạn trong nhóm đã cùng nhau hợp tác, hỗ trợ lẫn nhau trong suốt quá trình làm việc, cũng như cảm ơn gia đình và bạn bè đã luôn động viên, khích lệ tinh thần để chúng em có thể hoàn thành nhiệm vụ.

Mặc dù đã có nhiều cố gắng, nhưng do kiến thức và kinh nghiệm thực tế còn hạn chế nên đồ án chắc chắn không tránh khỏi thiếu sót. Nhóm chúng em rất mong nhận được sự góp ý của thầy cô và bạn bè để đề tài được hoàn thiện hơn.

Xin trân trọng cảm ơn!

# DANH MỤC BẢNG BIỂU

[2.5. Yêu cầu về phần cứng và phần mềm 17](#_Toc217517609)

[2.5.1 Yêu cầu tối thiểu 17](#_Toc217517610)

[2.5.2 Yêu cầu đề nghị 17](#_Toc217517611)

[3.2. Chức năng của hệ thống 18](#_Toc217517615)

[3.2.1 Guest functions (Chức năng dành cho khách vãng lai) 18](#_Toc217517616)

[3.2.2 Member functions (Chức năng dành cho thành viên) 19](#_Toc217517617)

[3.2.3 Admin functions (Chức năng dành cho quản trị viên) 21](#_Toc217517618)

[3.4. Database Design (Thiết kế cơ sở dữ liệu) 29](#_Toc217517635)

[3.4.1 Bảng users (người dùng) 29](#_Toc217517636)

[3.4.2 Bảng tenants (Người thuê phòng) 29](#_Toc217517637)

[3.4.3 Bảng hostels (Khu nhà trọ) 30](#_Toc217517638)

[3.4.4 Bảng rooms (Phòng trọ) 30](#_Toc217517639)

[3.4.5 Bảng amenities (Tiện nghi) 31](#_Toc217517640)

[3.4.6 Bảng room\_amenities (Tiện nghi của phòng) 31](#_Toc217517641)

[3.4.7 Bảng contracts (Hợp đồng thuê) 31](#_Toc217517642)

[3.4.8 Bảng contract\_members (Thành viên hợp đồng) 32](#_Toc217517643)

[3.4.9 Bảng services (Dịch vụ) 32](#_Toc217517644)

[3.4.10 Bảng meter\_readings (Chỉ số điện nước) 32](#_Toc217517645)

[3.4.11 Bảng bills (Hóa đơn) 33](#_Toc217517646)

[3.4.12 Bảng bill\_items (Chi tiết hóa đơn) 33](#_Toc217517647)

[3.4.13 Bảng payments (Thanh toán) 34](#_Toc217517648)

[3.4.14 Bảng booking\_requests (Yêu cầu xem phòng) 34](#_Toc217517649)

[3.4.15 Bảng feedbacks (Phản ánh / bảo trì) 34](#_Toc217517650)

[3.5 Table RelationShip Diagram (Mô hình quan hệ bảng) 35](#_Toc217517651)

[3.6 SiteMap (Sơ đồ cấu trúc site) 36](#_Toc217517652)

[3.6.1 Site map (Cấu trúc trang): Dành cho khách vãng lai 36](#_Toc217517653)

[3.6.2 Sitemap (Cấu trúc trang): Dành cho thành viên 36](#_Toc217517654)

[3.6.3 Dành cho quản trị viên 37](#_Toc217517655)

# DANH MỤC HÌNH ẢNH

[3.3. Systeme Designs (Thiết kế hệ thống) 24](#_Toc217517619)

[3.3.1 Thực thể users (người dùng) 24](#_Toc217517620)

[3.3.2 Thực thể tenants (Người thuê phòng) 24](#_Toc217517621)

[3.3.3 Thực thể hostels (Khu nhà trọ) 24](#_Toc217517622)

[3.3.4 Thực thể rooms (Phòng trọ) 25](#_Toc217517623)

[3.3.5 Thực thể amenities (Tiện nghi) 25](#_Toc217517624)

[3.3.6 Thực thể room\_amenities (Tiện nghi của phòng) 25](#_Toc217517625)

[3.3.7 Thực thể contracts (Hợp đồng thuê) 26](#_Toc217517626)

[3.3.8 Thực thể contract\_members (Thành viên hợp đồng) 26](#_Toc217517627)

[3.3.9 Thực thể services (Dịch vụ) 26](#_Toc217517628)

[3.3.10 Thực thể meter\_readings (Chỉ số điện) 27](#_Toc217517629)

[3.3.11 Thực thể bills (Hóa đơn) 27](#_Toc217517630)

[3.3.12 Thực thể bill\_items (Chi tiết hóa đơn) 27](#_Toc217517631)

[3.3.13 Thực thể payments (Thanh toán) 27](#_Toc217517632)

[3.3.14 Thực thể booking\_requests (Yêu cầu xem phòng) 28](#_Toc217517633)

[3.3.15 Thực thể feedbacks (Phản ánh / bảo trì) 28](#_Toc217517634)

[4.2 Một số giao diện người dùng và chức năng cụ thể 45](#_Toc217517669)

[4.2.1 Giao diện trang chủ 45](#_Toc217517670)

[4.2.2 Giao diện tra cứu hoá đơn theo phòng 46](#_Toc217517671)

[4.2.3 Giao diện thanh toán hoá đơn 46](#_Toc217517672)

[4.2.4 Giao diện gửi phản ánh 47](#_Toc217517673)

[4.2.5 Giao diện đăng nhập 47](#_Toc217517674)

[4.2.6 Giao diện quản trị viên 48](#_Toc217517675)

# CHƯƠNG 1: NGHIÊN CỨU CƠ SỞ LÝ THUYẾT

## Tổng quan về lập trình thiết kế giao diện website

### Giới thiệu HTML5

**HTML5** là phiên bản mới nhất của HyperText Markup Language (HTML) – ngôn ngữ đánh dấu siêu văn bản được sử dụng để xây dựng cấu trúc và nội dung của trang web. HTML5 được phát triển bởi W3C (World Wide Web Consortium) và WHATWG (Web Hypertext Application Technology Working Group) nhằm thay thế các chuẩn HTML/XHTML cũ, cung cấp nền tảng cho các ứng dụng web hiện đại với tính tương thích cao, đa nền tảng và đa thiết bị.

Khác với HTML truyền thống chỉ tập trung vào việc hiển thị nội dung, HTML5 được thiết kế với mục tiêu hỗ trợ các ứng dụng web phong phú (rich web applications), giảm sự phụ thuộc vào các plugin bên ngoài như Adobe Flash, đồng thời tối ưu trải nghiệm người dùng.

***Một số đặc điểm chính của HTML5***:

* Thẻ ngữ nghĩa (Semantic Elements): Bổ sung các thẻ như <header>, <footer>, <section>, <article>, <nav> giúp mô tả cấu trúc nội dung rõ ràng, hỗ trợ SEO và truy cập cho người khuyết tật.
* Hỗ trợ đa phương tiện: Tích hợp sẵn thẻ <audio> và <video> để phát nhạc, video mà không cần cài đặt plugin ngoài.
* Đồ họa và hiệu ứng: Cung cấp thẻ <canvas> và hỗ trợ SVG (Scalable Vector Graphics) để vẽ đồ họa, game, biểu đồ trực tiếp trên trình duyệt.
* Biểu mẫu nâng cao: Bổ sung nhiều loại input mới (email, date, number, range, color...), hỗ trợ kiểm tra dữ liệu (validation) phía client.
* Lưu trữ cục bộ: Hỗ trợ localStorage, sessionStorage, và IndexedDB, cho phép lưu dữ liệu offline trên trình duyệt.
* API mạnh mẽ: Bao gồm Geolocation API (xác định vị trí), Web Workers (xử lý song song), WebSocket (giao tiếp thời gian thực), Drag-and-Drop API…

### 1.1.2 Giới thiệu về CSS3

CSS3 (Cascading Style Sheets level 3) là phiên bản mới nhất của ngôn ngữ định kiểu, được sử dụng để thiết kế và trình bày nội dung cho các phần tử HTML trên trang web. CSS3 đóng vai trò quan trọng trong việc xây dựng giao diện trực quan, giúp tách biệt phần nội dung (HTML) với phần trình bày, từ đó nâng cao khả năng bảo trì và mở rộng của ứng dụng web.

So với các phiên bản trước, CSS3 được tổ chức thành nhiều module, mỗi module bổ sung các chức năng chuyên biệt. Điều này giúp các nhà phát triển dễ dàng áp dụng, cập nhật và tận dụng các tính năng mới mà không phải chờ toàn bộ tiêu chuẩn hoàn thiện.

Một số đặc điểm chính của CSS3:

* **Hiệu ứng động và hoạt ảnh**: Hỗ trợ transition, transform và animation giúp tạo chuyển động mượt mà mà không cần JavaScript.
* **Thiết kế đáp ứng (Responsive Design)**: Media Queries cho phép thay đổi giao diện tùy theo kích thước màn hình, tối ưu cho cả PC, tablet và smartphone.
* **Kiểu chữ đa dạng**: Tích hợp @font-face để sử dụng font chữ tùy chỉnh, không phụ thuộc vào bộ font cài sẵn trên thiết bị.
* **Màu sắc và nền nâng cao**: Hỗ trợ gradient, màu RGBA/HSLA và khả năng sử dụng nhiều lớp nền (multiple backgrounds).
* **Bố cục linh hoạt**: Cung cấp Flexbox và Grid Layout, giúp xây dựng bố cục hiện đại, dễ quản lý và phù hợp với xu hướng thiết kế web hiện nay.
* **Hiệu ứng trực quan**: border-radius, box-shadow, text-shadow giúp tạo giao diện mềm mại và trực quan hơn.

Bên cạnh CSS3 truyền thống, **Tailwind CSS** là một framework CSS theo hướng *utility-first*, giúp xây dựng giao diện web nhanh chóng và nhất quán thông qua việc sử dụng các lớp tiện ích có sẵn. Thay vì viết các đoạn CSS tùy chỉnh dài dòng, Tailwind CSS cho phép áp dụng trực tiếp các lớp định kiểu ngay trong thẻ HTML.

Tailwind CSS tập trung vào việc cung cấp các class nhỏ, đơn nhiệm như màu sắc, khoảng cách, kích thước, căn chỉnh, bố cục,… Nhờ đó, nhà phát triển có thể dễ dàng kiểm soát giao diện mà không cần tạo quá nhiều file CSS riêng biệt, đồng thời giảm thiểu tình trạng trùng lặp mã nguồn.

Một số đặc điểm nổi bật của Tailwind CSS:

* **Tiết kiệm thời gian phát triển**: Giảm đáng kể việc viết CSS thủ công, tăng tốc độ xây dựng giao diện.
* **Thiết kế đáp ứng linh hoạt**: Hỗ trợ sẵn các breakpoint cho responsive design như sm, md, lg, xl.
* **Dễ bảo trì và mở rộng**: Giao diện được xây dựng từ các utility class rõ ràng, hạn chế xung đột CSS.
* **Tùy biến cao**: Cho phép cấu hình theme (màu sắc, font chữ, spacing…) thông qua file cấu hình.
* **Phù hợp với các framework Back-end**: Dễ dàng tích hợp với Spring Boot và Thymeleaf trong các ứng dụng web hiện đại.

Nhờ những ưu điểm trên, Tailwind CSS ngày càng được sử dụng phổ biến trong các dự án web, đặc biệt là các ứng dụng yêu cầu giao diện hiện đại, nhất quán và khả năng phát triển nhanh.

### 1.1.3 Tổng quan về ngôn ngữ lập trình Javascript

JavaScript là ngôn ngữ lập trình kịch bản phía client, được dùng để tạo tính năng động và tương tác trên trang web. Nếu HTML xây dựng khung nội dung và CSS định dạng giao diện, thì JavaScript đóng vai trò "làm sống" trang web bằng cách xử lý các hành vi và sự kiện của người dùng.

Ban đầu, JavaScript được thiết kế để chạy trong trình duyệt nhằm thao tác với DOM (Document Object Model). Ngày nay, JavaScript đã phát triển thành một hệ sinh thái rộng lớn, không chỉ dùng trên client-side mà còn mở rộng sang server-side thông qua môi trường như Node.js.

Một số đặc điểm chính của JavaScript:

* **Ngôn ngữ thông dịch**: Không cần biên dịch, mã JavaScript được thực thi trực tiếp trong trình duyệt.
* **Lập trình hướng sự kiện**: Có thể lắng nghe và xử lý các sự kiện như click chuột, nhập dữ liệu, di chuyển chuột.
* **Thao tác DOM**: Cho phép thêm, xóa, thay đổi nội dung và cấu trúc HTML động.
* **Tương tác mạng**: Hỗ trợ AJAX để giao tiếp với server mà không cần tải lại trang.
* **Đa nền tảng**: Hoạt động trên hầu hết trình duyệt và hệ điều hành.
* **Hệ sinh thái phong phú**: Có hàng ngàn thư viện và framework như React, Angular, Vue hỗ trợ phát triển ứng dụng web hiện đại.

### 1.1.4 Thymeleaf Template Engine

Thymeleaf là một **Template Engine phía máy chủ** (Server-side Template Engine), được sử dụng phổ biến trong các ứng dụng Java, đặc biệt là khi kết hợp với **Spring Boot**. Thymeleaf cho phép xây dựng giao diện web động bằng cách kết hợp dữ liệu từ phía máy chủ với các trang HTML, từ đó hiển thị nội dung phù hợp cho người dùng.

Một trong những đặc điểm nổi bật của Thymeleaf là khả năng sử dụng **HTML thuần** làm template. Các tệp giao diện Thymeleaf vẫn là HTML hợp lệ, có thể mở trực tiếp trên trình duyệt mà không cần chạy ứng dụng, giúp quá trình thiết kế và chỉnh sửa giao diện trở nên thuận tiện hơn.

Thymeleaf cung cấp hệ thống **thuộc tính mở rộng** (ví dụ: th:text, th:if, th:each, …) để xử lý các chức năng như hiển thị dữ liệu, điều kiện rẽ nhánh, lặp danh sách và liên kết tài nguyên. Nhờ đó, việc truyền dữ liệu từ Controller đến View được thực hiện một cách rõ ràng và dễ kiểm soát.

Một số ưu điểm chính của Thymeleaf:

* **Dễ tích hợp với Spring Boot**: Được Spring hỗ trợ tốt, cấu hình đơn giản.
* **Tách biệt rõ ràng giữa giao diện và xử lý nghiệp vụ**: Giúp mã nguồn dễ đọc, dễ bảo trì.
* **Hỗ trợ xây dựng giao diện động**: Hiển thị dữ liệu theo điều kiện và vòng lặp.
* **Thân thiện với Front-end**: Template là HTML hợp lệ, dễ chỉnh sửa giao diện.
* **Phù hợp cho mô hình MVC**: Đóng vai trò là tầng View trong kiến trúc MVC.

Với những đặc điểm trên, Thymeleaf là lựa chọn phù hợp để xây dựng giao diện cho các ứng dụng web sử dụng Spring Boot, đáp ứng tốt yêu cầu về tính linh hoạt, khả năng mở rộng và bảo trì của hệ thống.

## Tổng quan về hệ quản trị cơ sở đữ liệu MySQL

Trong Trong các hệ thống ứng dụng web hiện nay, cơ sở dữ liệu đóng vai trò quan trọng trong việc lưu trữ, quản lý và truy xuất thông tin. Một hệ quản trị cơ sở dữ liệu (Database Management System – DBMS) hiệu quả sẽ giúp hệ thống hoạt động ổn định, đảm bảo tính toàn vẹn và an toàn của dữ liệu. Trong số các hệ quản trị cơ sở dữ liệu phổ biến hiện nay, **MySQL** là một trong những hệ quản trị được sử dụng rộng rãi, đặc biệt trong các ứng dụng web.

MySQL là hệ quản trị cơ sở dữ liệu quan hệ, sử dụng ngôn ngữ truy vấn SQL (Structured Query Language) để thao tác với dữ liệu. MySQL thường được sử dụng kết hợp với các công nghệ web như PHP, Java, Python và đặc biệt phù hợp khi tích hợp với **Spring Boot** trong các ứng dụng Java. Nhờ hiệu năng tốt, tính ổn định cao và dễ sử dụng, MySQL trở thành lựa chọn phổ biến cho nhiều dự án từ quy mô nhỏ đến lớn.

### 1.2.1 Giới thiệu về MySQL

MySQL là một **hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở**, được phát triển ban đầu bởi công ty MySQL AB và hiện nay thuộc quyền sở hữu của Oracle Corporation. MySQL hoạt động theo mô hình **Client – Server**, trong đó máy chủ MySQL chịu trách nhiệm quản lý dữ liệu và xử lý các yêu cầu truy vấn từ phía ứng dụng.

MySQL lưu trữ dữ liệu dưới dạng các **bảng (table)** có mối quan hệ với nhau thông qua khóa chính (Primary Key) và khóa ngoại (Foreign Key). Cách tổ chức này giúp dữ liệu được quản lý một cách logic, giảm thiểu dư thừa và đảm bảo tính nhất quán. MySQL hỗ trợ đầy đủ các thao tác cơ bản như thêm, sửa, xóa và truy vấn dữ liệu thông qua ngôn ngữ SQL.

Bên cạnh đó, MySQL còn hỗ trợ nhiều **engine lưu trữ** khác nhau, trong đó phổ biến nhất là **InnoDB**, cho phép hỗ trợ giao dịch (transaction), khóa ngoại và đảm bảo tính toàn vẹn dữ liệu. Điều này giúp MySQL phù hợp với các hệ thống yêu cầu độ chính xác và an toàn cao.

Trong các ứng dụng web sử dụng Java và Spring Boot, MySQL thường được kết nối thông qua **Spring Data JPA** hoặc JDBC, giúp việc thao tác với cơ sở dữ liệu trở nên đơn giản và hiệu quả hơn. Nhờ khả năng tích hợp tốt và cộng đồng hỗ trợ lớn, MySQL được đánh giá là hệ quản trị cơ sở dữ liệu linh hoạt và đáng tin cậy.

### 1.2.2 Ưu điểm

MySQL sở hữu nhiều ưu điểm nổi bật, góp phần làm cho hệ quản trị này được sử dụng rộng rãi trong thực tế:

* **Mã nguồn mở và miễn phí**: MySQL cho phép người dùng sử dụng và triển khai mà không tốn chi phí bản quyền, phù hợp cho sinh viên và các dự án học tập. **Tính bảo mật mạnh mẽ**: Hỗ trợ mã hóa dữ liệu, phân quyền người dùng chi tiết.
* **Hiệu năng cao và ổn định**: MySQL có khả năng xử lý nhanh các truy vấn dữ liệu, đáp ứng tốt cho các ứng dụng web có lượng người dùng lớn. **Hỗ trợ đa dạng**: Làm việc tốt với nhiều ngôn ngữ lập trình và nền tảng, đặc biệt là .NET.
* **Dễ sử dụng và triển khai**: Việc cài đặt, cấu hình và quản lý MySQL tương đối đơn giản, phù hợp với cả người mới bắt đầu.
* **Hỗ trợ đa nền tảng**: MySQL có thể hoạt động trên nhiều hệ điều hành khác nhau như Windows, Linux và macOS.
* **Khả năng mở rộng tốt**: MySQL có thể sử dụng cho cả các hệ thống nhỏ lẫn các hệ thống lớn với lượng dữ liệu lớn.
* **Tích hợp tốt với các framework**: MySQL hoạt động hiệu quả khi kết hợp với Spring Boot, Hibernate và Spring Data JPA.

### 1.2.3 Nhược điểm

Bên cạnh các ưu điểm, MySQL vẫn tồn tại một số hạn chế nhất định:

* **Hạn chế ở các tính năng nâng cao**: So với một số hệ quản trị cơ sở dữ liệu khác như Oracle hay PostgreSQL, MySQL còn thiếu một số tính năng chuyên sâu cho các hệ thống lớn. **Yêu cầu tài nguyên hệ thống lớn**: Cần cấu hình phần cứng tương đối mạnh để đạt hiệu suất tối ưu.
* **Khả năng xử lý dữ liệu phức tạp chưa tối ưu**: Với các truy vấn phức tạp hoặc hệ thống yêu cầu xử lý dữ liệu lớn, MySQL có thể gặp khó khăn về hiệu năng.
* **Phụ thuộc vào engine lưu trữ**: Một số tính năng quan trọng chỉ hoạt động tốt khi sử dụng engine InnoDB.
* **Khả năng mở rộng theo chiều ngang còn hạn chế**: Việc mở rộng hệ thống MySQL ở quy mô rất lớn đòi hỏi cấu hình và giải pháp bổ sung.

Mặc dù còn tồn tại một số nhược điểm, nhưng với các ứng dụng web sử dụng Spring Boot trong phạm vi đồ án và thực tế phổ biến, MySQL vẫn đáp ứng tốt các yêu cầu về lưu trữ, quản lý và truy xuất dữ liệu.

## Tổng quan về ngôn ngữ lập trình phía máy chủ

### Giới thiệu ngôn ngữ lập trình Java

Java là một **ngôn ngữ lập trình hướng đối tượng**, được phát triển bởi Sun Microsystems vào năm 1995 và hiện nay thuộc sở hữu của Oracle Corporation. Java được thiết kế với mục tiêu xây dựng các ứng dụng có tính **độc lập nền tảng cao**, an toàn và dễ mở rộng. Nhờ những đặc điểm này, Java đã và đang được sử dụng rộng rãi trong nhiều lĩnh vực như phát triển ứng dụng web, ứng dụng doanh nghiệp, ứng dụng di động và các hệ thống phần mềm lớn. ASP.NET Core là một phần của .NET, được thiết kế đặc biệt cho phát triển ứng dụng web và dịch vụ web hiện đại. Nó hỗ trợ kiến trúc module hóa, hiệu suất cao, khả năng mở rộng tốt và có thể chạy trên nhiều hệ điều hành khác nhau.

Một trong những đặc trưng quan trọng nhất của Java là nguyên lý **“Write Once, Run Anywhere”**, cho phép chương trình Java có thể chạy trên nhiều hệ điều hành khác nhau mà không cần chỉnh sửa mã nguồn. Điều này đạt được nhờ cơ chế biên dịch mã nguồn Java thành **bytecode**, sau đó được thực thi bởi **Máy ảo Java (Java Virtual Machine – JVM)**. JVM đóng vai trò trung gian, giúp chương trình Java hoạt động thống nhất trên các nền tảng khác nhau như Windows, Linux hay macOS.

Java hỗ trợ mạnh mẽ mô hình **lập trình hướng đối tượng**, với các khái niệm cốt lõi như lớp (class), đối tượng (object), kế thừa (inheritance), đa hình (polymorphism) và đóng gói (encapsulation). Những đặc điểm này giúp việc tổ chức và quản lý mã nguồn trở nên rõ ràng, dễ bảo trì và thuận tiện cho việc mở rộng hệ thống trong tương lai.

Bên cạnh đó, Java còn được đánh giá cao về **tính an toàn và ổn định**. Ngôn ngữ này cung cấp cơ chế quản lý bộ nhớ tự động thông qua **Garbage Collection**, giúp hạn chế các lỗi liên quan đến rò rỉ bộ nhớ. Ngoài ra, Java có hệ thống kiểm soát truy cập và xử lý ngoại lệ chặt chẽ, góp phần nâng cao độ tin cậy của ứng dụng.

Java sở hữu một **hệ sinh thái thư viện và framework phong phú**, hỗ trợ hiệu quả cho quá trình phát triển phần mềm. Trong lĩnh vực phát triển ứng dụng web, Java thường được sử dụng kết hợp với các framework như **Spring Boot**, **Hibernate** và **Spring Data JPA**, giúp đơn giản hóa việc xây dựng các ứng dụng theo mô hình MVC và kết nối với cơ sở dữ liệu.

Với những ưu điểm về tính linh hoạt, khả năng mở rộng và cộng đồng hỗ trợ lớn, Java là lựa chọn phù hợp cho việc phát triển các ứng dụng web hiện đại. Trong đồ án này, Java được sử dụng làm **ngôn ngữ lập trình phía máy chủ**, đóng vai trò xử lý nghiệp vụ và kết nối với cơ sở dữ liệu trong hệ thống.

### 1.3.2 Giới thiệu Spring Boot Framework (Spring MVC, Spring Data JPA, Spring Security)

Spring Boot là framework phát triển ứng dụng Java dựa trên Spring Framework, được thiết kế nhằm **đơn giản hóa quá trình cấu hình và triển khai** ứng dụng. Spring Boot cho phép xây dựng nhanh các ứng dụng web theo kiến trúc **Model – View – Controller (MVC)**, giúp tách biệt rõ ràng giữa giao diện, xử lý nghiệp vụ và dữ liệu.

Spring Boot bao gồm các thành phần chính:

* **Spring MVC**: Đảm nhiệm việc tiếp nhận và xử lý các yêu cầu từ phía người dùng, điều phối luồng xử lý giữa Model và View. Spring MVC hỗ trợ xây dựng Controller, ánh xạ URL và trả về giao diện hoặc dữ liệu tương ứng.
* **Spring Data JPA** Hỗ trợ thao tác với cơ sở dữ liệu thông qua mô hình ánh xạ đối tượng – quan hệ (ORM). Spring Data JPA giúp giảm thiểu việc viết câu lệnh SQL thủ công, tăng tính nhất quán và dễ bảo trì mã nguồn.
* **Spring Security**: Cung cấp các cơ chế xác thực và phân quyền người dùng, giúp bảo vệ ứng dụng trước các truy cập trái phép. Spring Security hỗ trợ kiểm soát quyền truy cập vào từng chức năng trong hệ thống.

# CHƯƠNG 2: TỔNG QUAN VỀ ĐỀ TÀI

## Đặt vấn đề

Trong những năm gần đây, cùng với quá trình đô thị hóa và phát triển kinh tế – xã hội, nhu cầu thuê phòng trọ tại các thành phố lớn và khu công nghiệp ngày càng gia tăng. Đối tượng thuê phòng trọ chủ yếu là sinh viên, người lao động, công nhân và người có thu nhập trung bình – thấp.

Tuy nhiên, công tác quản lý phòng trọ tại nhiều khu nhà trọ hiện nay vẫn còn thực hiện theo phương pháp thủ công, sử dụng sổ sách hoặc các file Excel rời rạc. Cách quản lý này gây ra nhiều hạn chế như khó kiểm soát thông tin phòng, khách thuê, hợp đồng, chi phí điện nước và tình trạng thanh toán.

Do đó, việc xây dựng **hệ thống quản lý phòng trọ** là cần thiết nhằm tin học hóa quy trình quản lý, giúp chủ nhà trọ theo dõi thông tin một cách chính xác, nhanh chóng, giảm sai sót và nâng cao hiệu quả quản lý.

## Hệ thống hiện tại

Hiện nay, tại nhiều khu phòng trọ, việc quản lý vẫn chủ yếu được thực hiện bằng phương pháp thủ công, cụ thể:

* Quản lý thông tin phòng trọ bằng sổ ghi chép hoặc file Excel đơn giản..
* Khó quản lý thông tin khách hàng.
* Tốn nhiều thời gian cho nhân viên.
* Không hỗ trợ đặt phòng trực tuyến cho khách hàng.

Những hạn chế trên khiến công tác quản lý trở nên kém hiệu quả, đặc biệt khi số lượng phòng và khách thuê ngày càng tăng.

## Hệ thống đề xuất

Hệ thống đề xuất là **hệ thống quản lý phòng trọ trên nền tảng web/ứng dụng**, nhằm hỗ trợ chủ nhà trọ trong việc quản lý toàn bộ hoạt động cho thuê phòng một cách hiệu quả.

Các chức năng chính của hệ thống bao gồm:

* Quản lý thông tin phòng trọ (loại phòng, giá phòng, trạng thái phòng)..
* Quản lý thông tin khách thuê phòng.
* Quản lý hợp đồng thuê phòng.
* Quản lý chi phí điện, nước và các dịch vụ liên quan.
* Theo dõi tình trạng thanh toán tiền phòng.
* Thống kê, báo cáo doanh thu theo tháng, năm.

Hệ thống giúp tự động hóa các quy trình quản lý, giảm thiểu sai sót, tiết kiệm thời gian và nâng cao trải nghiệm cho cả chủ nhà trọ và người thuê.

## Giới hạn của hệ thống

Do giới hạn về thời gian, nguồn lực và phạm vi thực hiện đề tài, hệ thống quản lý phòng trọ được xây dựng với một số giới hạn như sau:

* Chưa hỗ trợ thanh toán trực tuyến.
* Chưa tích hợp các chức năng quản lý nâng cao như quản lý nhiều khu trọ khác nhau.
* Chỉ phục vụ cho đối tượng chủ nhà trọ quy mô nhỏ và vừa.
* Chưa hỗ trợ ứng dụng trên nhiều nền tảng khác nhau.

Trong tương lai, hệ thống có thể được mở rộng và nâng cấp thêm các chức năng mới nhằm đáp ứng tốt hơn nhu cầu thực tế và nâng cao hiệu quả quản lý.

## Yêu cầu về phần cứng và phần mềm

### 2.5.1 Yêu cầu tối thiểu

|  |  |  |
| --- | --- | --- |
|  | **Server (Máy chủ)** | **Client (Máy khách)** |
| Hardware | \* Ram: 4GB.  \* HDD >=20GB trống.  \* CPU: Intel Core i3 thế hệ 6 \* Kết nối Internet. | \* Ram: 2GB.  \* HDD >=5GB trống  \* CPU :Intel Core i3 thế hệ 5  \* Kết nối Internet. |
| Software | \* Hệ điều hành: Windows 10/11 hoặc Linux  \* Java Development Kit (JDK) 21  **\*** MySQL Sever 8.0+  **\*** Maven 3.6+ | \* Hệ điều hành: Windows 10/11, macOS, Linux.  \* Trình duyệt: Chrome 100+, FireFox 90+, Egde 100+. |

### 2.5.2 Yêu cầu đề nghị

|  |  |  |
| --- | --- | --- |
|  | **Server** | **Client** |
| Hardware | \* Ram: 8GB trở lên.  \* SSD: >= 50GB.  \* CPU: Intel Core i5 thế hệ 8 trở lên hoặc AMD Ryzen 5  \* Kết nối Internet tốc độ cao. | \* Ram: 4Gb trở lên.  \* SSD: >= 10GB  \* CPU: Intel Core i5 thế hệ 7.  \* Kết nối Internet ổn định. |
| Software | \* Hệ điều hành: Windows 11 hoặc Unbutu 22.04 LTS.  \* Java Development Kit (JDK) 21  \* SQL Server 2022 Express trở lên.  **\*** MySQL Sever 8.0+  **\*** Maven 3.9+  \* Intellij Idea hoặc Eclipse IDE | \* Windows 11, macOS Ventura+, Unbutu 22.04+.  \* Trình duyệt: Chrome 120+, FireFox 120+, Egde 120+. (Phiên bản mới nhất) |

# CHƯƠNG 3:

# PHÂN TÍCH YÊU CẦU KHÁCH HÀNG VÀ THIẾT KẾ HỆ THỐNG

## Người dùng của hệ thống

Đối tượng người dùng của hệ thống bao gồm

* Guest (Khách vãng lai)
* Member (Thành viên - Khách đăng kí tài khoản của trang web )
* Admin (Quản trị hệ thống).

## Chức năng của hệ thống

### 3.2.1 Guest functions (Chức năng dành cho khách vãng lai)

* **Xem phòng**

|  |  |
| --- | --- |
| **Mô tả** | Khách hàng có thể xem thông tin chi tiết từng loại sản phẩm như tên phòng, nột thất, hình ảnh phòng, giá cả… |
| **Nhập** | Lựa chọn loại phòng cần xem. |
| **Xử lý** | Tìm kiếm trong CSDL. |
| **Xuất** | Hiển thị thông tin mà khách hàng muốn xem. |

* **Tìm kiếm phòng**

|  |  |
| --- | --- |
| **Mô tả** | Khách hàng tìm kiếm sản phẩm mà họ cần |
| **Nhập** | Nhập tên của sản phẩm cần xem. |
| **Xử lý** | Tìm kiếm trong CSDL. |
| **Xuất** | Hiển thị thông tin về sản phầm mà khách hàng tìm kiếm. |

* **Đăng ký thành viên**

|  |  |
| --- | --- |
| **Mô tả** | Khách hàng đăng kí tài khoản thành viên của website. |
| **Nhập** | Nhập thông tin của khách hàng. |
| **Xử lý** | Lưu thông tin khách hàng vào CSDL. |
| **Xuất** | Chuyển sang trang chủ đăng nhập để khách hàng đăng nhập tài khoản vừa đăng kí |

### Member functions (Chức năng dành cho thành viên)

Thành viên chính thức của website sẽ có các chức năng giống như khách vãng lai như:

* + Xem phòng.
  + Tìm kiếm phòng.
  + Lựa chọn phòng.

***Ngoài ra thành viên chính thức còn có các chức năng sau***:

* **Đăng nhập**

|  |  |
| --- | --- |
| **Mô tả** | Đăng nhập vào website bằng cách nhập username và password đã đăng ký. |
| **Nhập** | Nhập username và mật khẩu đã đăng ký. |
| **Xử lý** | Truy cập kiểm tra xem tài khoản có tồn tại hay không. |
| **Xuất** | Hiển thị thông báo đăng nhập thành công khi khách hàng nhập đúng thông tin đã đăng ký và quay về trang chủ. Thông báo sai thông tin đăng nhập nếu tài khoản chưa đăng ký, yêu cầu nhập lại thông tin đăng nhập! |

* **Đăng xuất**

|  |  |
| --- | --- |
| **Mô tả** | Đăng xuất khỏi website. |
| **Nhập** | Nhấn nút thoát để đăng xuất tài khoản. |
| **Xử lý** | * Hủy toàn bộ session của người dùng hiện tại * Đưa trở về chức năng của khách vãng lai. |
| **Xuất** | Hiển thị giao diện như khách vãng lai sau khi đã đăng xuất khỏi hệ thống. |

* **Phiếu đặt phòng**

|  |  |
| --- | --- |
| **Mô tả** | Phiếu đặt phòng được tạo sau khi yêu cầu đặt phòng của khách hàng được xác nhận, dùng để ghi nhận thông tin lưu trú chính thức. |
| **Nhập** | Thông tin khách hàng, thông tin phòng/loại phòng, ngày nhận phòng, ngày trả phòng, số lượng phòng, số lượng khách. |
| **Xử lý** | Xác nhận yêu cầu đặt phòng, tạo phiếu đặt phòng trong hệ thống, gán mã phiếu đặt phòng, cập nhật trạng thái phòng. |
| **Xuất** | Thông báo đơn đặt phòng đang được xác nhận |

* **Yêu cầu đặt phòng**

|  |  |
| --- | --- |
| **Mô tả** | Yêu cầu đặt phòng của người dùng |
| **Nhập** | Thông tin khách hàng, phòng được chọn, ngày nhận và trả phòng, số lượng phòng, số lượng khách, tổng số tiền phải thanh toán, thao tác đơn hàng |
| **Xử lý** | Kiểm tra tình trạng phòng theo thời gian đặt, tính tổng chi phí dựa theo ngày lưu trú và loại phòng, cập nhật trạng thái phòng tương ướng |
| **Xuất** | Hiển thỉ thông tin chi tiết yêu cầu đặt phòng, thông báo kết quả đặt phòng |

* **Đánh giá**

|  |  |
| --- | --- |
| **Mô tả** | Đánh giá của khách hàng |
| **Nhập** | Khách hàng chọn số sao (từ 1 đến 5) và nhập nội dung nhận xét/bình luận. |
| **Xử lý** | Xác thực dữ liệu, sau đó lưu thông tin đánh giá vào cơ sở dữ liệu. |
| **Xuất** | Hiển thị đánh giá (số sao, nội dung bình luận, ngày đánh giá, tên khách hàng) trên trang chi tiết sản phẩm để người dùng khác tham khảo. |

### Admin functions (Chức năng dành cho quản trị viên)

***Quản trị viên có các chức năng chính như sau:***

* **Đăng nhập quản trị**

|  |  |
| --- | --- |
| **Mô tả** | Quản trị viên đăng nhập vào hệ thống bằng cách nhập username và password vào form đăng nhập. |
| **Nhập** | Nhập tài khoản và mật khẩu của quản trị viên. |
| **Xử lý** | Kiểm tra thông tin nhập vào có hợp lệ hay không. |
| **Xuất** | * Chuyển đến trang quản trị nếu đăng nhập thành công * Quay lại trang đăng nhập nếu nhập sai. |

* **Đăng xuất quản trị**

|  |  |
| --- | --- |
| **Mô tả** | Thoát khỏi tài khoản quản trị viên. |
| **Nhập** | Click vào nút thoát từ giao diện quản trị. |
| **Xử lý** | Hủy session quản trị viên đã đăng nhập. |
| **Xuất** | Quay lại trang chủ. |

* **Quản lý phòng trọ**

|  |  |
| --- | --- |
| **Mô tả** | Quản trị viên thực hiện quản lý thông tin khách hàng sử dụng hệ thống đặt phòng phòng trọ |
| **Nhập** | Thông tin khách hàng: họ tên, số điện thoại, email…  Thông tin tài khoản (nếu có): tên đăng nhập, trạng thái |
| **Xử lý** | Thực hiện các chức năng thêm, sửa, xóa, tìm kiếm khách hàng, kiểm tra tính hợp lệ của dữ liệu nhập vào, lưu và cập nhật thông tin khách hàng vào cơ sở dữ liệu. |
| **Xuất** | Hiển thị danh sách khách hàng, cập nhật thông tin khách hàng sau khi thao tác thành công. |

* **Quản lý tài khoản khách hàng/QTV khác**

|  |  |
| --- | --- |
| **Mô tả** | Quản trị viên có quyền xem thông tin hoặc khóa/kích hoạt, xóa các tài khoản của thành viên. |
| **Nhập** | Chọn lệnh tương ứng của tài khoản cần thao tác. |
| **Xử lý** | * Với việc xem thông tin tài khoản: * Hiển thị thông tin của khách hàng. * Đối với việc xóa tài khoản: * Thay đổi trạng thái quyền đăng nhập website. * Khóa/kích hoạt tài khoản: * Admin có quyền khóa/kích hoạt tài khoản của khách hàng. |
| **Xuất** | Hiện thông tin đã thay đổi. |

* **Quản lý phòng**

|  |  |
| --- | --- |
| **Mô tả** | Quản trị viên quản lý thông tin các phòng trong khách sạn và trạng thái sử dụng của phòng. |
| **Nhập** | Thông tin phòng: mã phòng, tên phòng, loại phòng, trạng thái phòng. |
| **Xử lý** | Thực hiện thêm, sửa, xóa phòng, kiểm tra tình trạng phòng trước khi cập nhật, cập nhật trạng thái phòng theo các phiếu đặt phòng. |
| **Xuất** | Hiển thị danh sách phòng, hiển thị trạng thái phòng hiện tại trong hệ thống. |

* **Quản lý loại bảng**

|  |  |
| --- | --- |
| **Mô tả** | Quản trị viên quản lý các loại phòng nhằm phân loại phòng theo đặc điểm và mức giá. |
| **Nhập** | Tên loại phòng, giá phòng, số lượng người tối đa, mô tả loại phòng |
| **Xử lý** | Thêm mới, chỉnh sửa, xóa loại phòng, kiểm tra dữ liệu hợp lệ (giá phòng > 0, số người hợp lý), cập nhật thông tin loại phòng vào cơ sở dữ liệu |
| **Xuất** | Hiển thị danh sách các loại phòng, cập nhật thông tin loại phòng sau khi thay đổi |

* **Quản lý đánh giá**

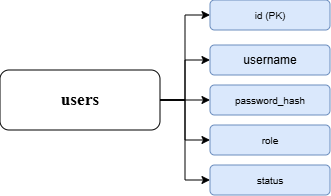
|  |  |
| --- | --- |
| **Mô tả** | Quản trị viên giám sát và kiểm duyệt các đánh giá sản phẩm của khách hàng. |
| **Nhập** | Các đánh giá từ khách hàng (số sao, nội dung bình luận, thông tin người đánh giá). |
| **Xử lý** | Hệ thống cho phép quản trị viên duyệt, ẩn, hoặc xóa các đánh giá vi phạm nội quy; lưu thay đổi vào cơ sở dữ liệu. |
| **Xuất** | Cập nhật danh sách đánh giá hiển thị trên website, chỉ giữ lại các đánh giá hợp lệ. |

## Systeme Designs (Thiết kế hệ thống)

***Mô tả chi tiết các thuộc tính của các thực thể***:

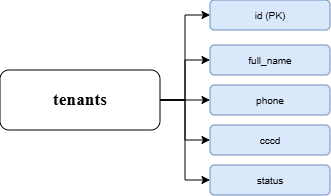
### 3.3.1 Thực thể users (người dùng)

Lưu trữ thông tin người dùng của hệ thống.



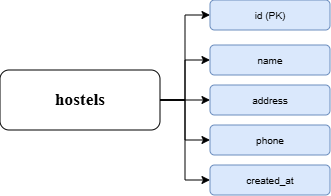
### 3.3.2 Thực thể tenants (Người thuê phòng)

Lưu trữ thông tin người thuê phòng của hệ thống phòng trọ.



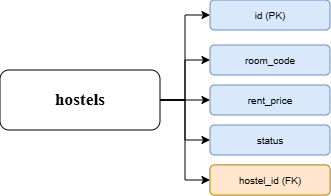
### 3.3.3 Thực thể hostels (Khu nhà trọ)

Lưu trữ khu nhà trọ cho thuê của hệ thống



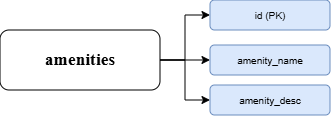
### 3.3.4 Thực thể rooms (Phòng trọ)

Lưu trữ đặt phòng trọ của hệ thống



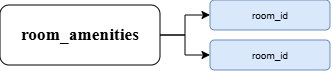
### 3.3.5 Thực thể amenities (Tiện nghi)

Lưu trữ các loại tiện nghi của hệ thống phòng trọ



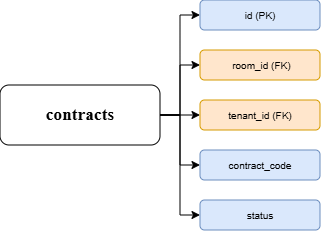
### 3.3.6 Thực thể room\_amenities (Tiện nghi của phòng)

Lưu trữ các loại tiện nghi của phòng trọ



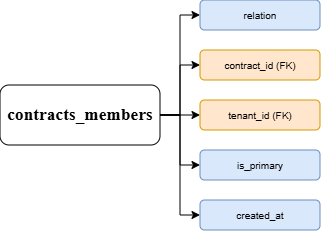
### 3.3.7 Thực thể contracts (Hợp đồng thuê)

Lưu trữ (các) hợp đồng thuê của người dùng



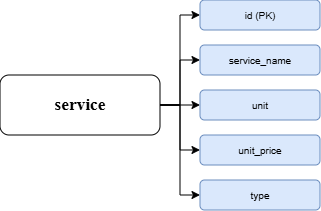
### 3.3.8 Thực thể contract\_members (Thành viên hợp đồng)

Lưu trữ những (các) người trong hợp dồng thuê của khách hàng



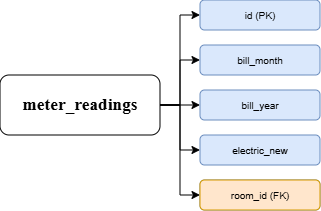
### 3.3.9 Thực thể services (Dịch vụ)

Lưu trữ dịch vụ của website



### 3.3.10 Thực thể meter\_readings (Chỉ số điện)

Lưu trữ chi tiết chỉ số điện của phòng



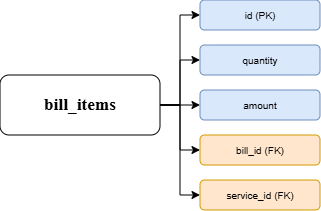
### 3.3.11 Thực thể bills (Hóa đơn)

Lưu trữ hoá đơn



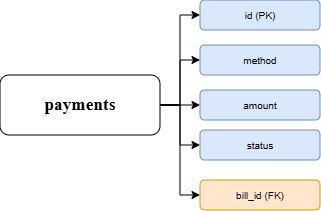
### 3.3.12 Thực thể bill\_items (Chi tiết hóa đơn)

Lưu trữ chi tiết hoá đơn



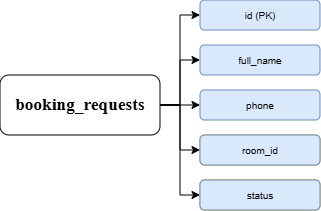
### 3.3.13 Thực thể payments (Thanh toán)

Lưu trữ chi tiết thanh toán



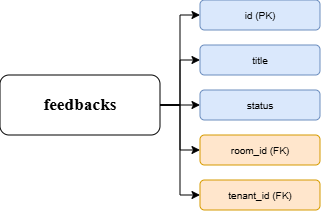
### 3.3.14 Thực thể booking\_requests (Yêu cầu xem phòng)

Lưu trữ chi tiết yêu cầu xem phòng



### 3.3.15 Thực thể feedbacks (Phản ánh / bảo trì)

Lưu trữ phản ánh của người dùng



## Database Design (Thiết kế cơ sở dữ liệu)

Thiết kế có sở dữ liệu trên hệ quản trị dữ liệu MySQL

### 3.4.1 Bảng users (người dùng)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên cột (Field)** | **Kiểu dữ liệu (DataType)** | **Null** | **Ràng buộc (Constraint)** | **Mô tả (Content)** |
| id | bigint | NO | |  | | --- | |  |  |  | | --- | | PK, AI | | |  | | --- | |  |  |  | | --- | | ID người dùng | |
| username | Varchar  (50) | NO |  | |  | | --- | |  |  |  | | --- | | Tên đăng nhập | |
| password\_hash | VARCHAR  (255) | NO | UNIQUE | Mật khẩu mã hóa |
| role | VARCHAR  (255) | NO |  | Vai trò admin/user |
| status | VARCHAR  (255) | NO |  | Trạng thái tài khoản |

### 3.4.2 Bảng tenants (Người thuê phòng)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên cột (Field**) | **Kiểu dữ liệu** | **Null** | **Ràng buộc** | **Mô tả** |
| id | bigint | NO | |  | | --- | |  |   PK, AI | ID người thuê |
| full\_name | varchar(100) | NO |  | Họ tên |
| phone | varchar(20) | NO | UNIQUE | Số điện thoại |
| cccd | varchar(20) | YES | UNIQUE | CCCD |
| status | |  | | --- | |  |   enum | NO |  | Trạng thái hoạt động |

### 3.4.3 Bảng hostels (Khu nhà trọ)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Null** | **Ràng buộc** | **Mô tả** |
| id | bigint | NO | PK, AI | ID khu trọ |
| name | Varchar  (120) | NO |  | Tên khu trọ |
| address | Varchar  (255) | NO |  | Địa chỉ |
| phone | Varchar  (30) | YES |  | SĐT liên hệ |
| created\_at | datetime | NO |  | Ngày tạo |

### 3.4.4 Bảng rooms (Phòng trọ)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên cột (Field**) | **Kiểu dữ liệu** | **Null** | **Ràng buộc** | **Mô tả** |
| id | bigint | NO | PK, AI | ID phòng |
| room\_code | Varchar  (20) | NO | PK, AI | Mã phòng |
| rent\_price | Decimal  (38,2) | NO |  | Giá thuê |
| status | enum | NO |  | EMPTY / RENTING |
| hostel\_id | bigint | YES | FK | Thuộc khu trọ |

### 3.4.5 Bảng amenities (Tiện nghi)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Null** | **Ràng buộc** | **Mô tả** |
| id | bigint | NO | |  | | --- | |  |  |  | | --- | | PK, AI | | ID tiện nghi |
| amenity\_name | varchar(100) | NO | UNIQUE | Tên tiện nghi |
| amenity\_desc | varchar(255) | YES |  | Mô tả |

### 3.4.6 Bảng room\_amenities (Tiện nghi của phòng)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Null** | **Ràng buộc** | **Mô tả** |
| room\_id | bigint | NO | PK, FK | Phòng |
| amenity\_id | bigint | NO | PK, FK | Tiện nghi |

### 3.4.7 Bảng contracts (Hợp đồng thuê)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Null** | **Ràng buộc** | **Mô tả** |
| id | bigint | NO | PK, AI | ID hợp đồng |
| contract\_code | varchar(30) | NO | UNIQUE | Mã hợp đồng |
| room\_id | bigint | NO | FK | Phòng thuê |
| tenant\_id | bigint | NO | FK | Người thuê |
| status | enum | NO |  | ACTIVE / ENDED |

### 3.4.8 Bảng contract\_members (Thành viên hợp đồng)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Null** | **Ràng buộc** | **Mô tả** |
| contract\_id | bigint | NO | PK, FK | Hợp đồng |
| tenant\_id | bigint | NO | PK, FK | Người ở |
| relation | varchar(50) | YES |  | Quan hệ |
| is\_primary | tinyint | NO |  | Đại diện |
| created\_at | datetime | NO |  | Ngày thêm |

### 3.4.9 Bảng services (Dịch vụ)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Null** | **Ràng buộc** | **Mô tả** |
| id | bigint | NO | PK, AI | ID dịch vụ |
| service\_name | Varchar(100) | NO | UNIQUE | Tên dịch vụ |
| unit | Varchar(20) | NO |  | Đơn vị |
| unit\_price | Decimal(12,2) | NO |  | Giá |
| type | enum | NO | DEFAULT 1 | FIXED/METER |

### 3.4.10 Bảng meter\_readings (Chỉ số điện nước)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên cột**) | **Kiểu dữ liệu** | **Null** | **Ràng buộc** | **Mô tả** |
| id | bigint | NO | PK, AI | ID |
| room\_id | bigint | NO | FK | Phòng |
| bill\_month | int | NO |  | Tháng |
| bill\_year | int | NO |  | Năm |
| electric\_new | int | NO |  | Số điện mới |

### 3.4.11 Bảng bills (Hóa đơn)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Null** | **Ràng buộc** | **Mô tả** |
| id | bigint | NO | PK, AI | ID hóa đơn |
| contract\_id | bigint | NO | FK | Hợp đồng |
| room\_id | |  | | --- | |  |  |  | | --- | | bigint | | NO | FK | |  | | --- | |  |  |  | | --- | | Phòng | |
| total\_amount | decimal(38,2) | NO |  | Tổng tiền |
| status | enum | NO |  | PAID / UNPAID |

### 3.4.12 Bảng bill\_items (Chi tiết hóa đơn)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Null** | **Ràng buộc** | **Mô tả** |
| id | bigint | NO | PK, AI | ID |
| bill\_id | bigint | NO | FK | Hóa đơn |
| service\_id | |  | | --- | |  |  |  | | --- | | bigint | | YES | FK | |  | | --- | |  |   Dịch vụ |
| quantity | decimal(12,2) | NO |  | Số lượng |
| amount | decimal(12,2) | NO |  | Thành tiền |

### 3.4.13 Bảng payments (Thanh toán)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Null** | **Ràng buộc** | **Mô tả** |
| id | bigint | NO | PK, AI | ID |
| bill\_id | bigint | NO | FK | Hóa đơn |
| amount | |  | | --- | |  |  |  | | --- | | decimal(38,2) | | NO |  | |  | | --- | |  |  |  | | --- | | Số tiền | |
| method | enum | NO |  | CASH / BANK |
| status | enum | NO |  | SUCCESS |

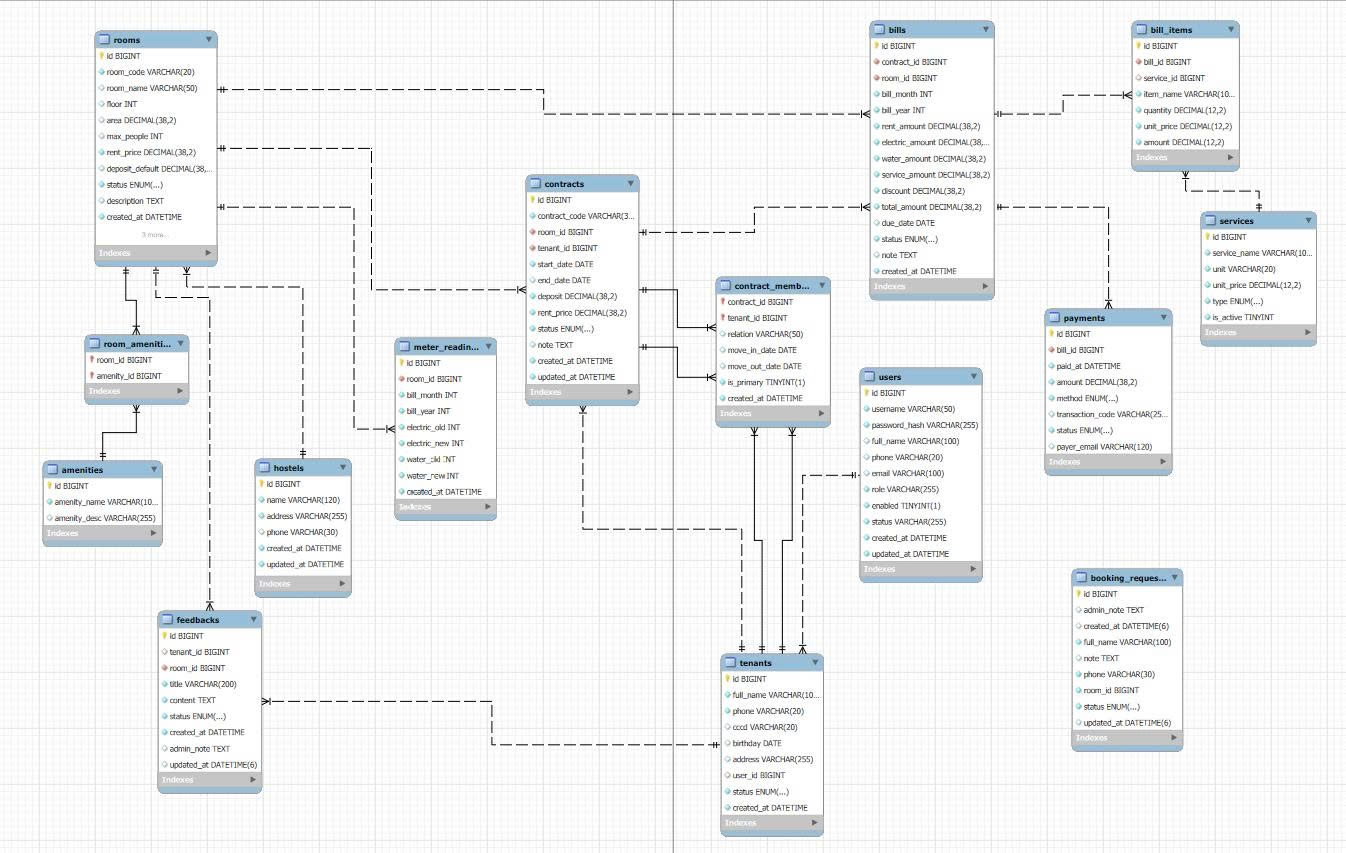
### 3.4.14 Bảng booking\_requests (Yêu cầu xem phòng)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Null** | **Ràng buộc** | **Mô tả** |
| id | bigint | No | PK, AI | ID |
| full\_name | varchar(100) | No |  | Người đặt |
| phone | |  | | --- | |  |  |  | | --- | | Varchar(30) | | No |  | |  | | --- | |  |  |  | | --- | | SĐT | |
| room\_id | VARCHAR(50) | No |  | Phòng |
| status | DATETIME | No |  | NEW / APPROVED |

### 3.4.15 Bảng feedbacks (Phản ánh / bảo trì)

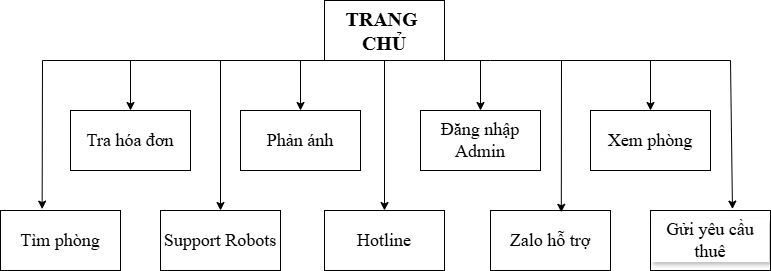
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Null** | **Ràng buộc** | **Mô tả** |
| id | bigint | NO | PK, AI | ID |
| tenant\_id | bigint | YES | FK | Người gửi |
| room\_id | |  | | --- | |  |  |  | | --- | | bigint | | NO | FK | |  | | --- | |  |  |  | | --- | | Phòng | |
| title | varchar(200) | NO |  | Tiêu đề |
| status | enum | NO |  | NEW / DONE |

## Table RelationShip Diagram (Mô hình quan hệ bảng)

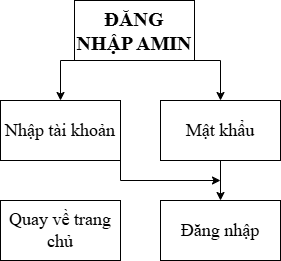


## 3.6 SiteMap (Sơ đồ cấu trúc site)

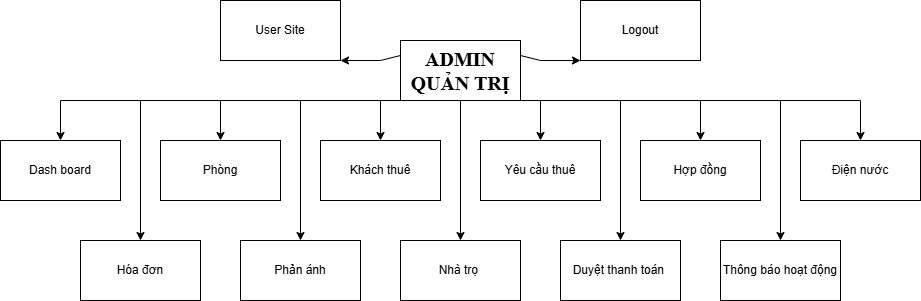
### 3.6.1 Site map (Cấu trúc trang): Dành cho khách vãng lai



### 3.6.2 Sitemap (Cấu trúc trang): Dành cho login admin

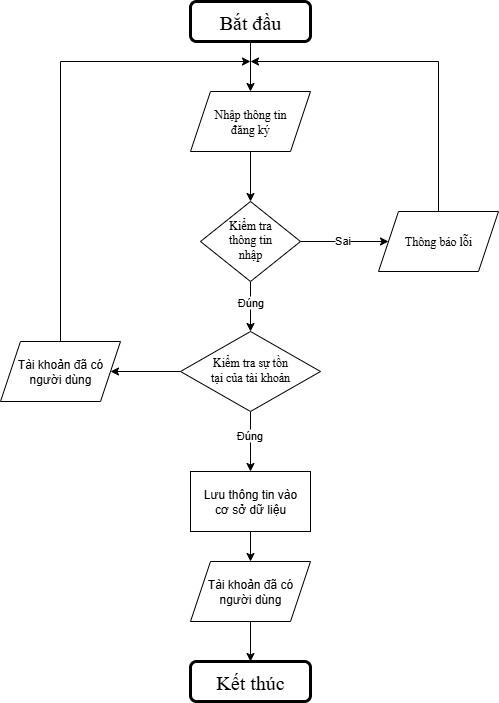


### Dành cho quản trị viên



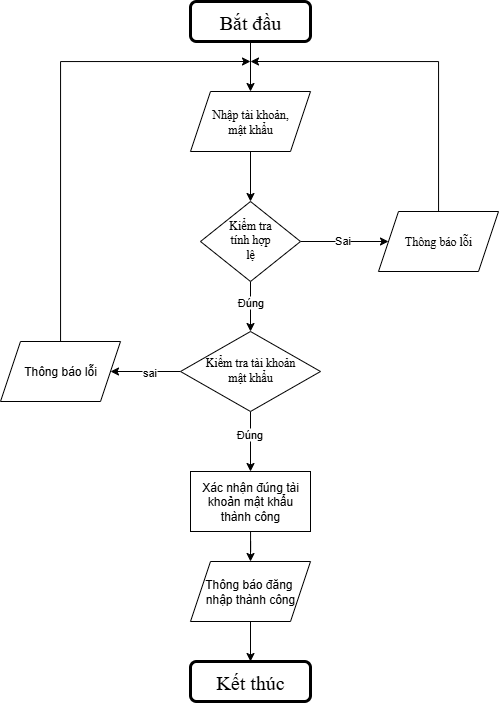
## 3.7 Algorithms (Giải thuật)

### 3.7.1 Đăng ký

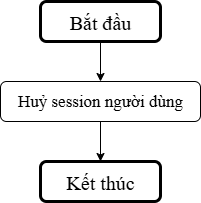


|  |  |
| --- | --- |
| * **Kiểm tra thông tin nhập**:   Kiểm tra các thông tin bắt buộc nhập theo ràng buộc khi thiết kế cơ sở dữ liệu:   * + Tài khoản, Mật khẩu, Email, Điện thoại,.. | * **Kiểm tra sự tồn tại của tài khoản:**   Tài khoản là duy nhất đối với mỗi khách hàng. Nếu đã có người dùng rồi thì không cho phép đăng ký. |

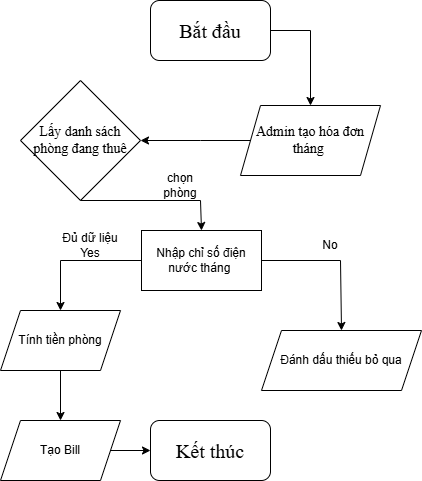
### 3.7.2 Đăng nhập



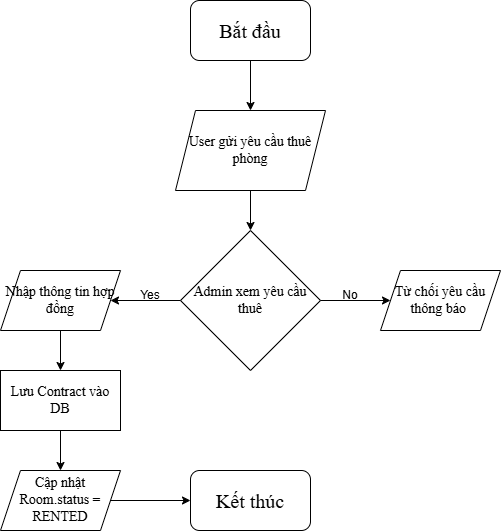
### 3.7.3 Đăng xuất



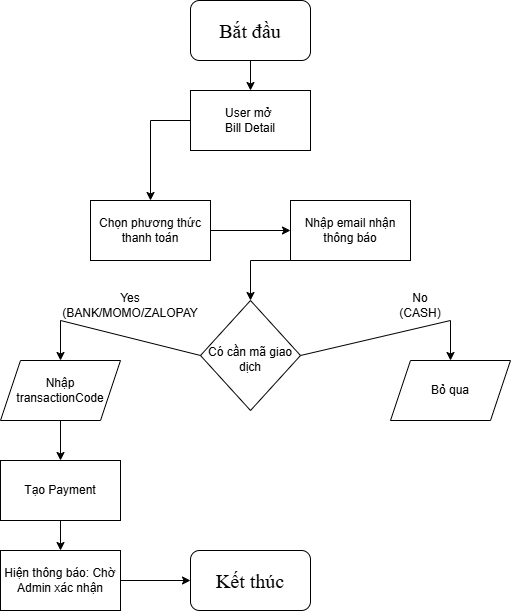
### 3.7.4 Tạo hóa đơn



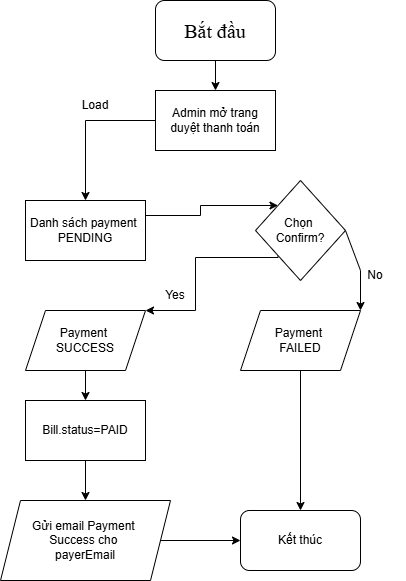
### 3.7.5 User gửi yêu cầu thuê đến Admin



### 3.7.6 User gửi yêu cầu thanh toán và chờ thông báo gmail



### 3.7.7 Admin duyệt thanh toán và gửi thông báo gmail



# CHƯƠNG 4: XÂY DỰNG HỆ THỐNG

## Xây dựng phần mềm

## Mã nguồn chương trình

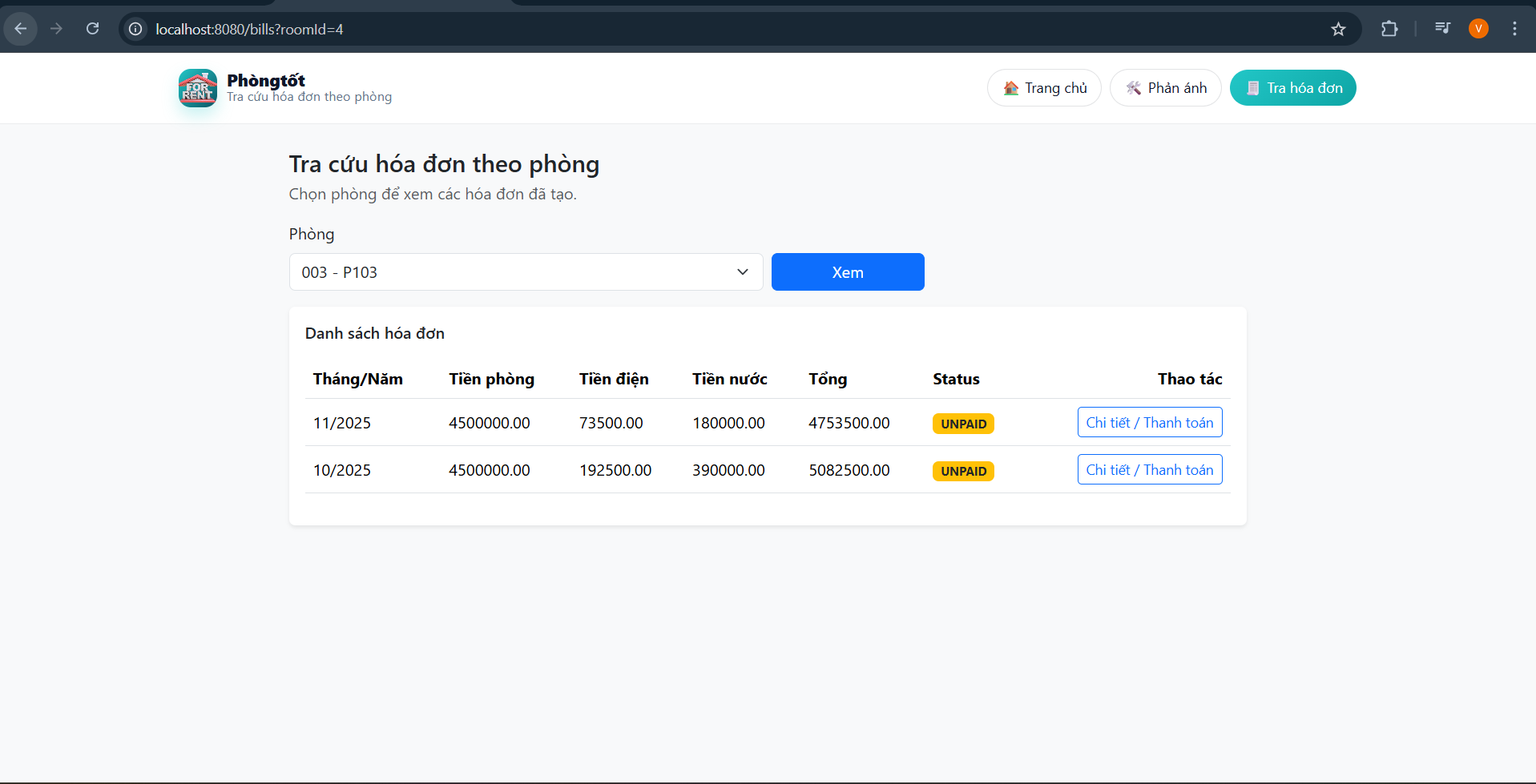
GITHUB

## 4.2 Một số giao diện người dùng và chức năng cụ thể

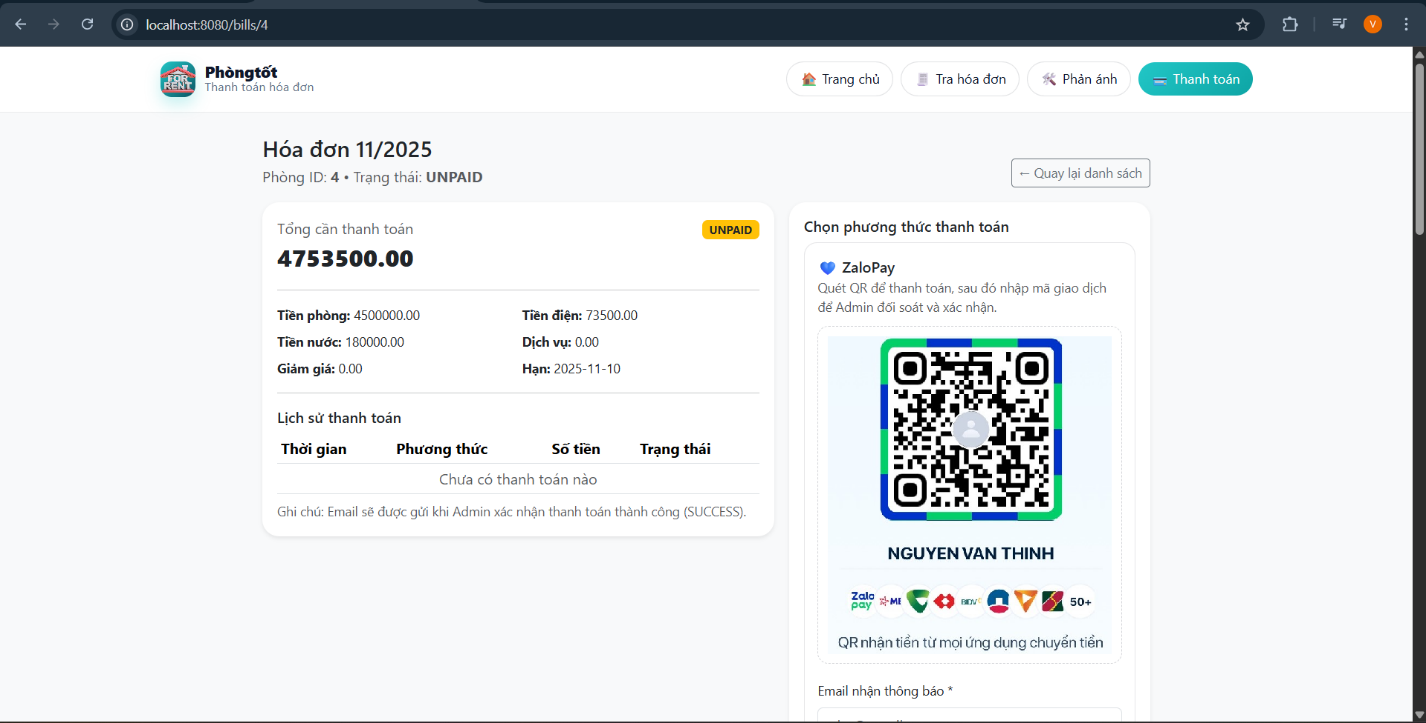
### 4.2.1 Giao diện trang chủ



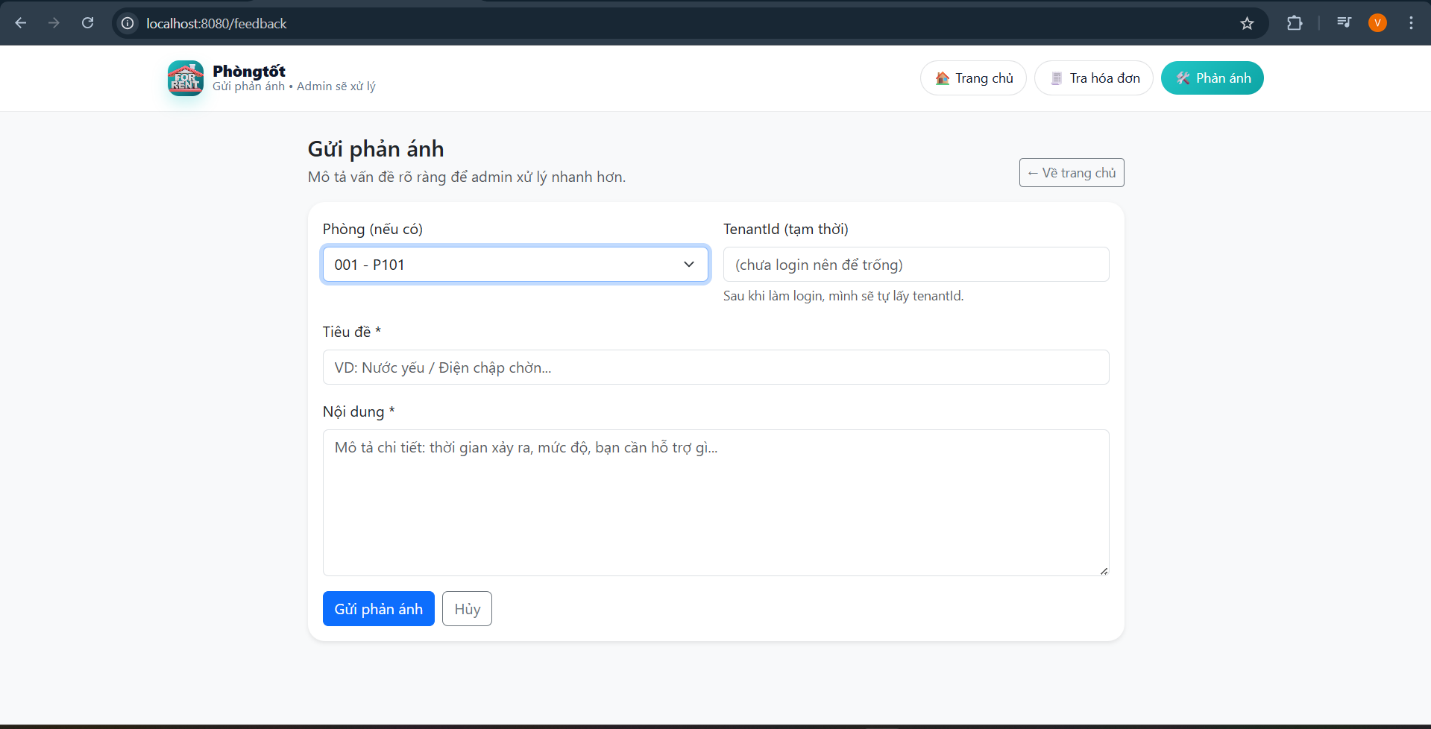
### 4.2.2 Giao diện tra cứu hoá đơn theo phòng



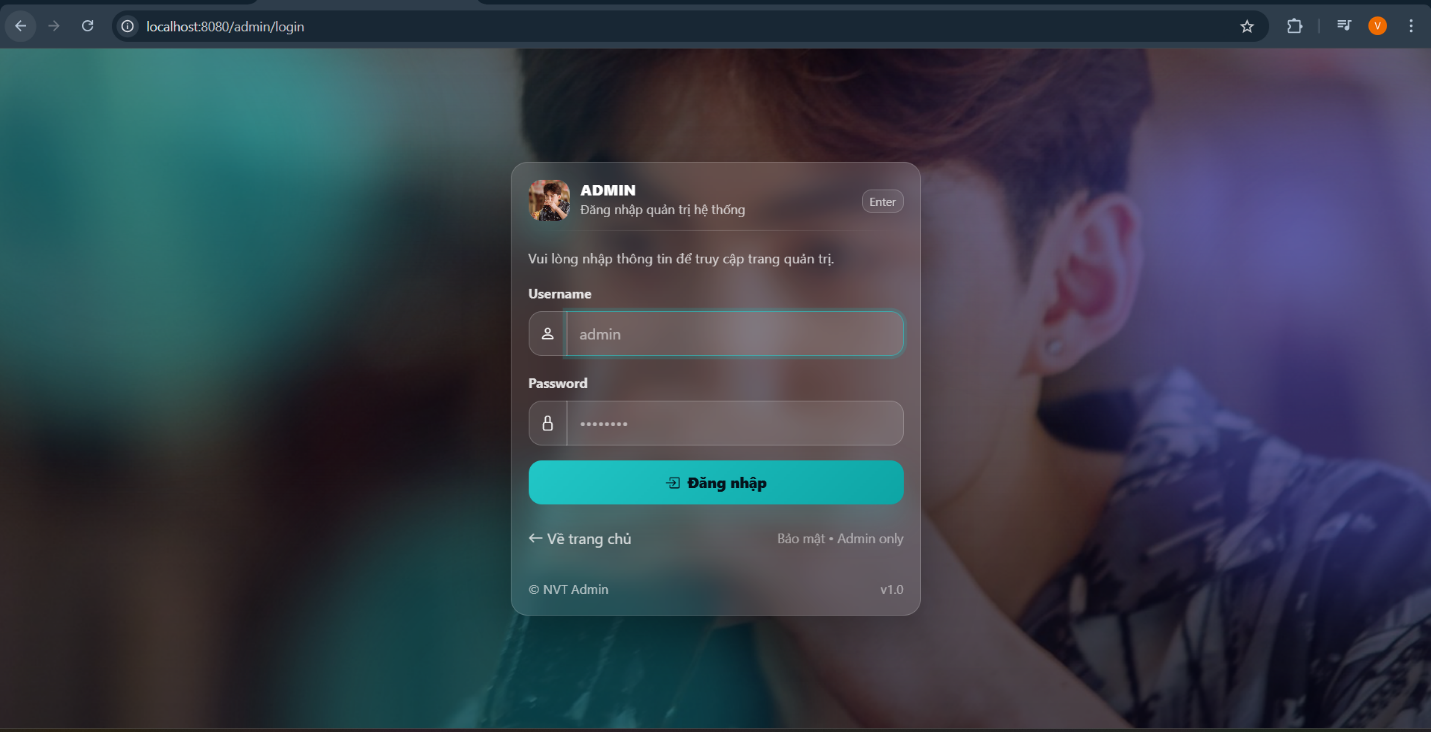
### 4.2.3 Giao diện thanh toán hoá đơn



### 4.2.4 Giao diện gửi phản ánh

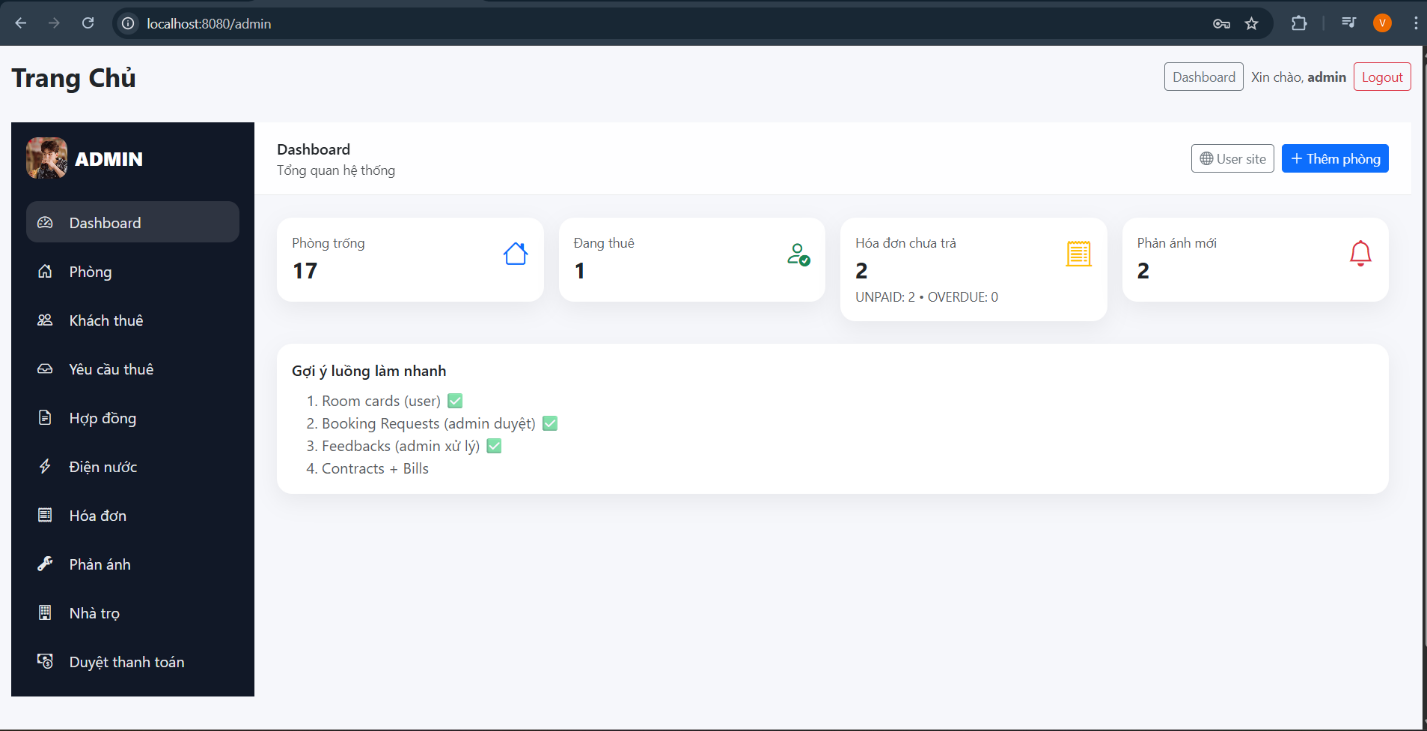


### 4.2.5 Giao diện đăng nhập

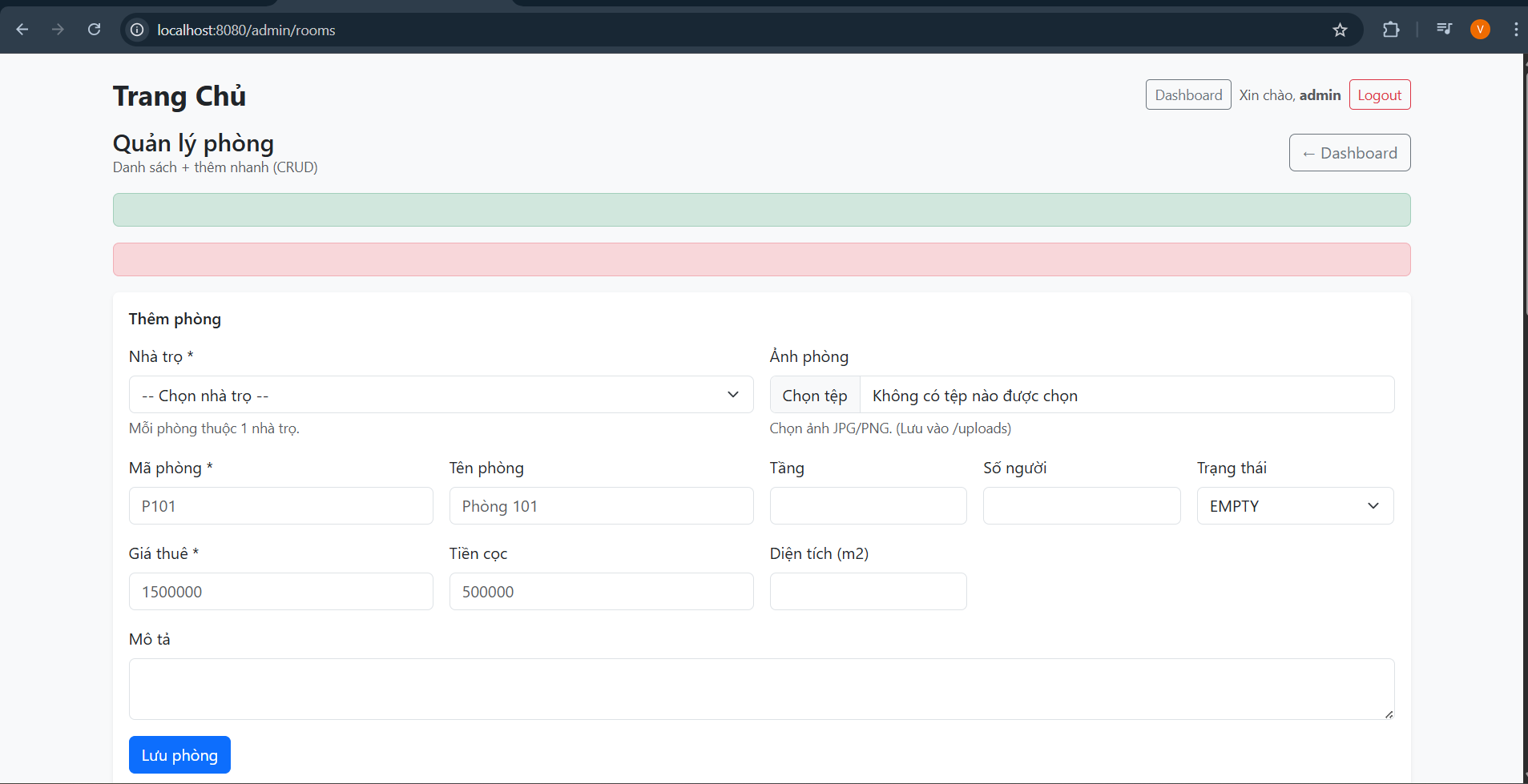


### 4.2.6 Giao diện quản trị viên

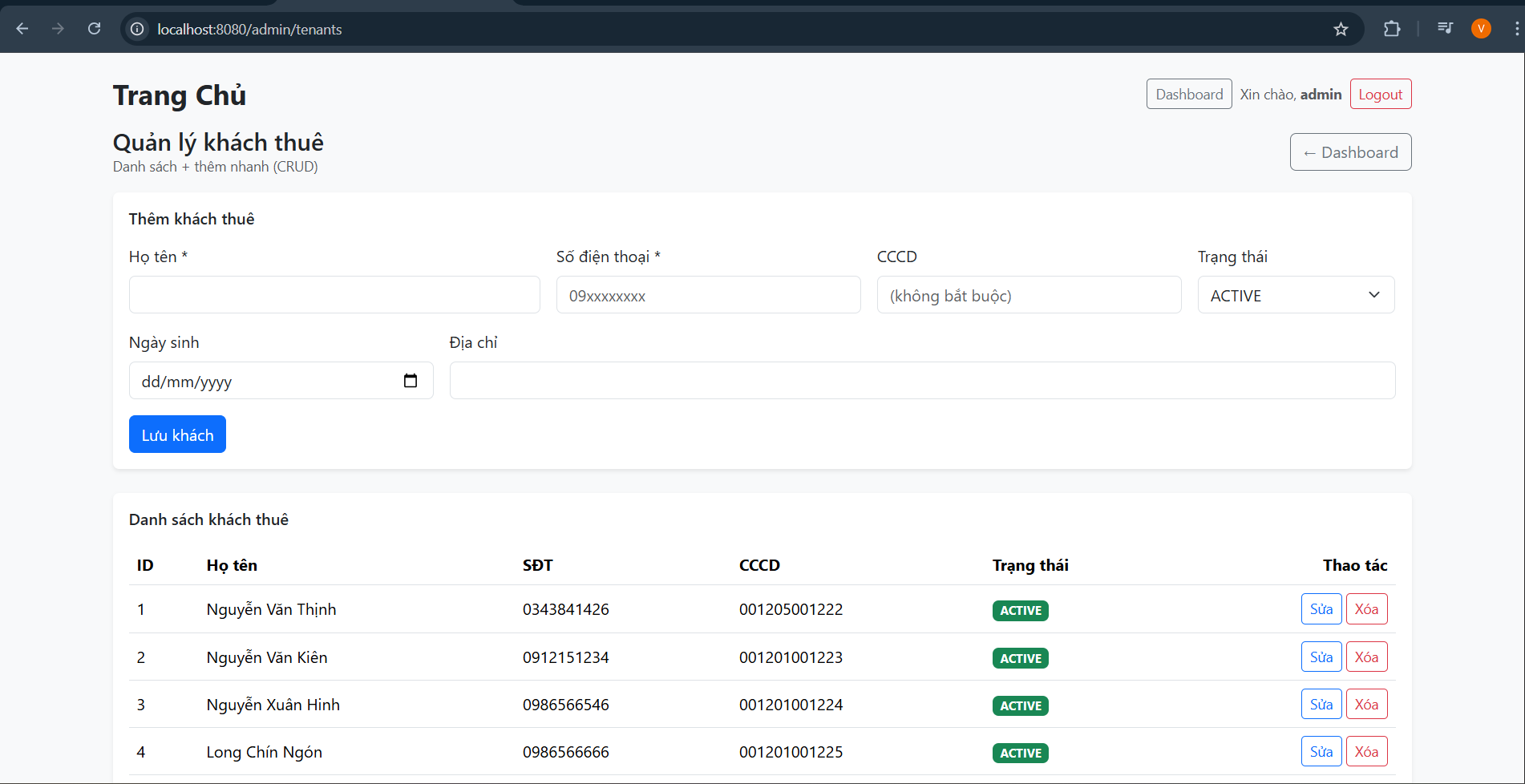
Trang chủ Admin



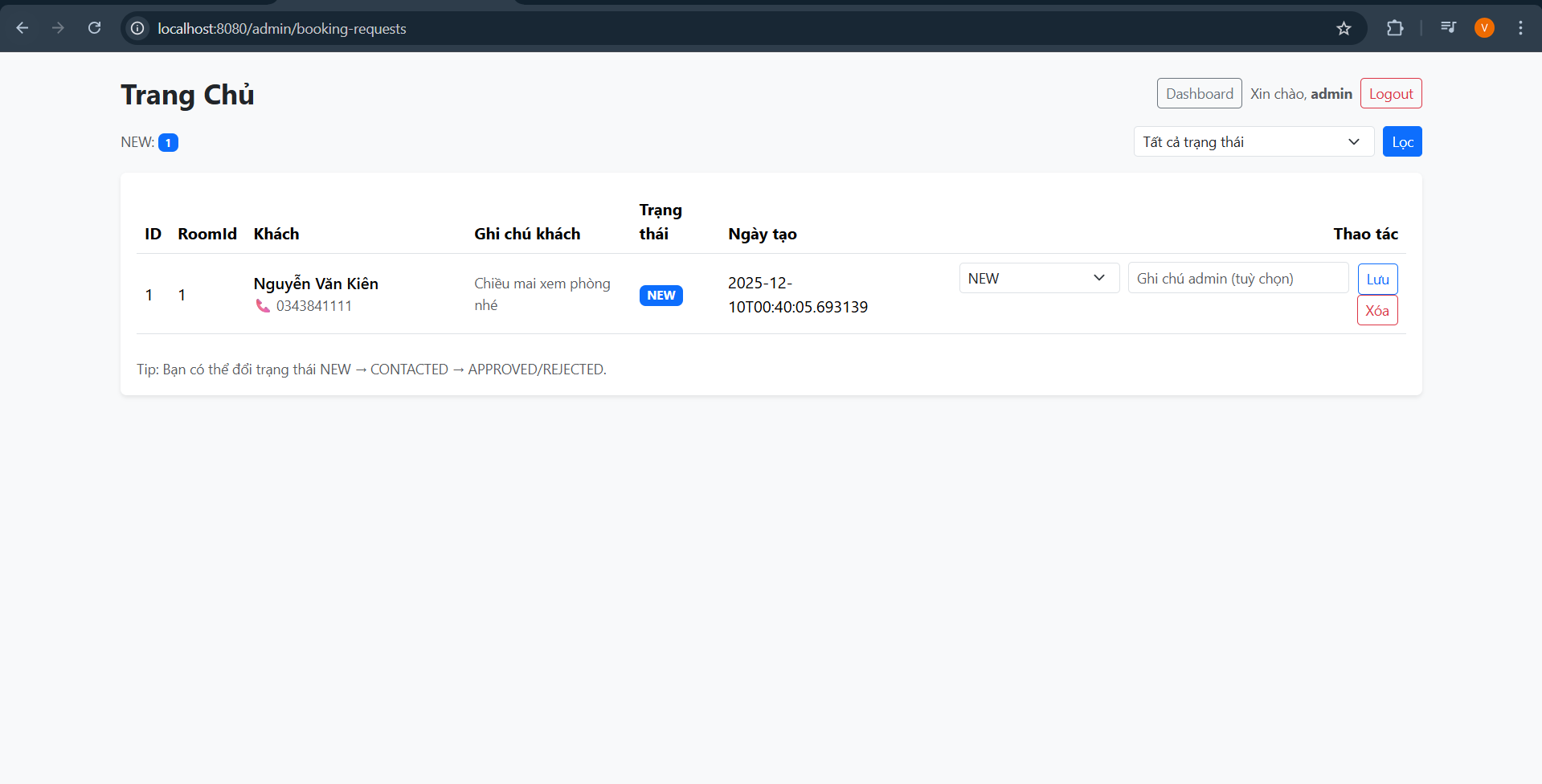
Quản lý phòng



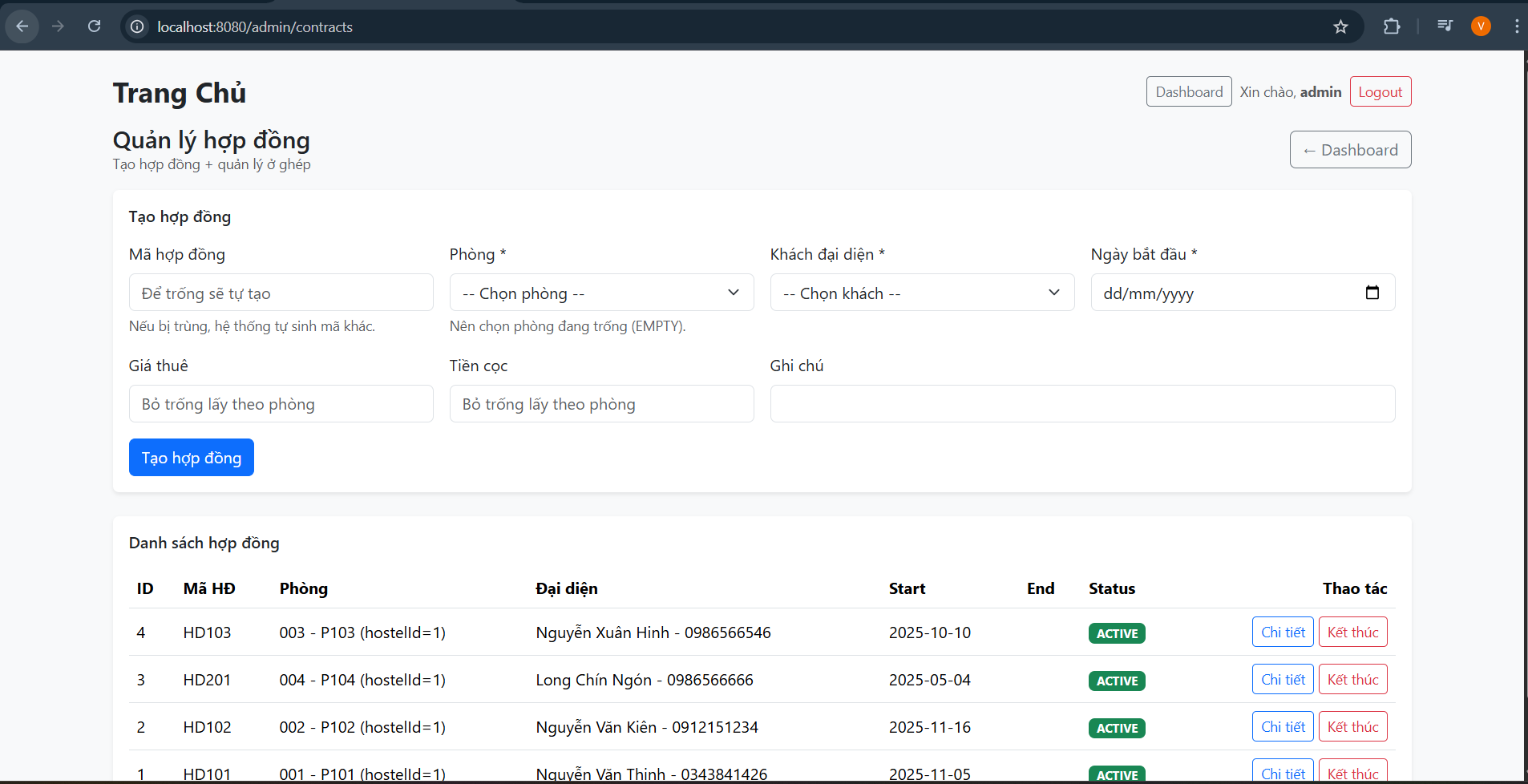
Danh sách khách thuê



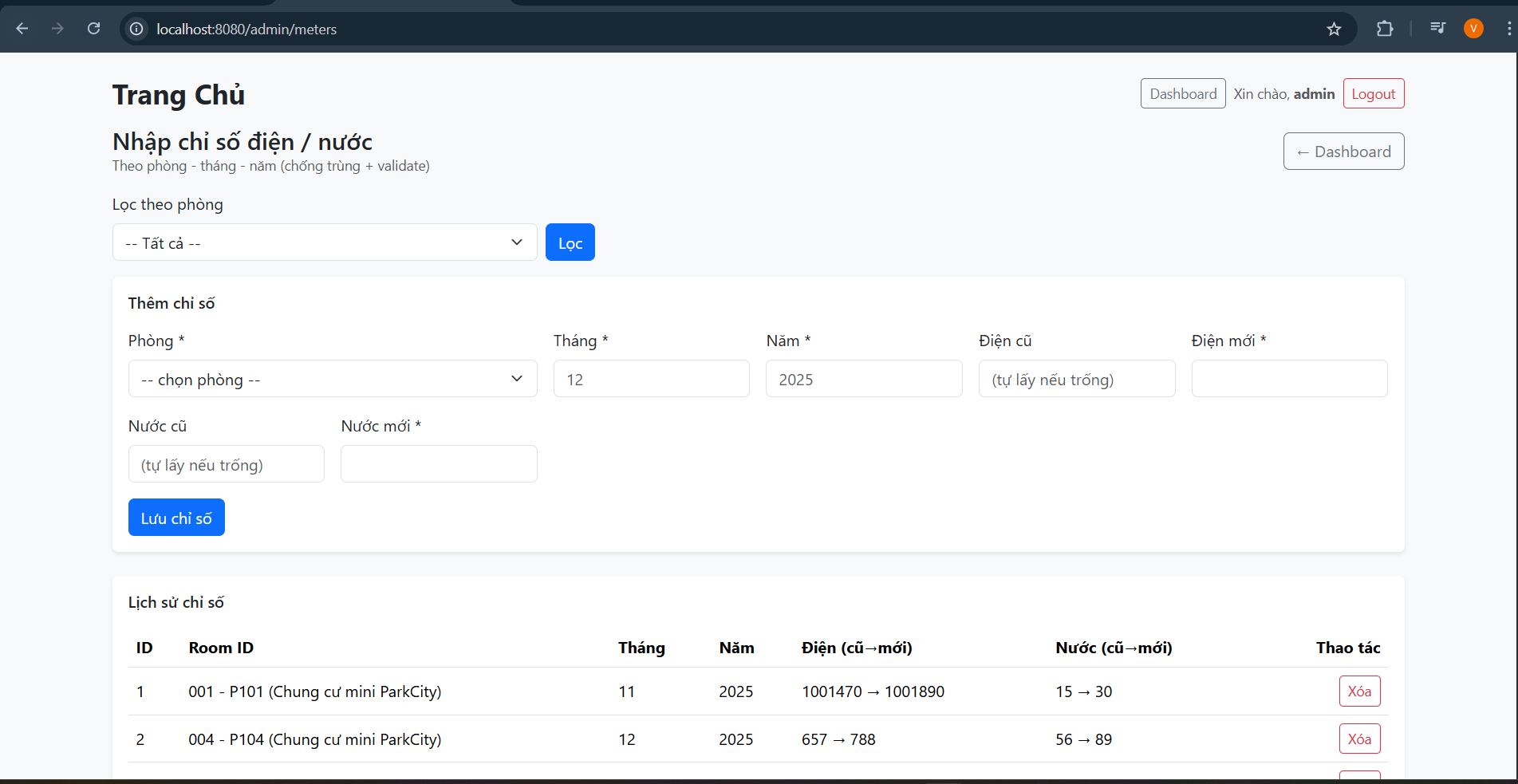
Giao diện Booking Request (hẹn xem phòng)



Quản lý hợp đồng



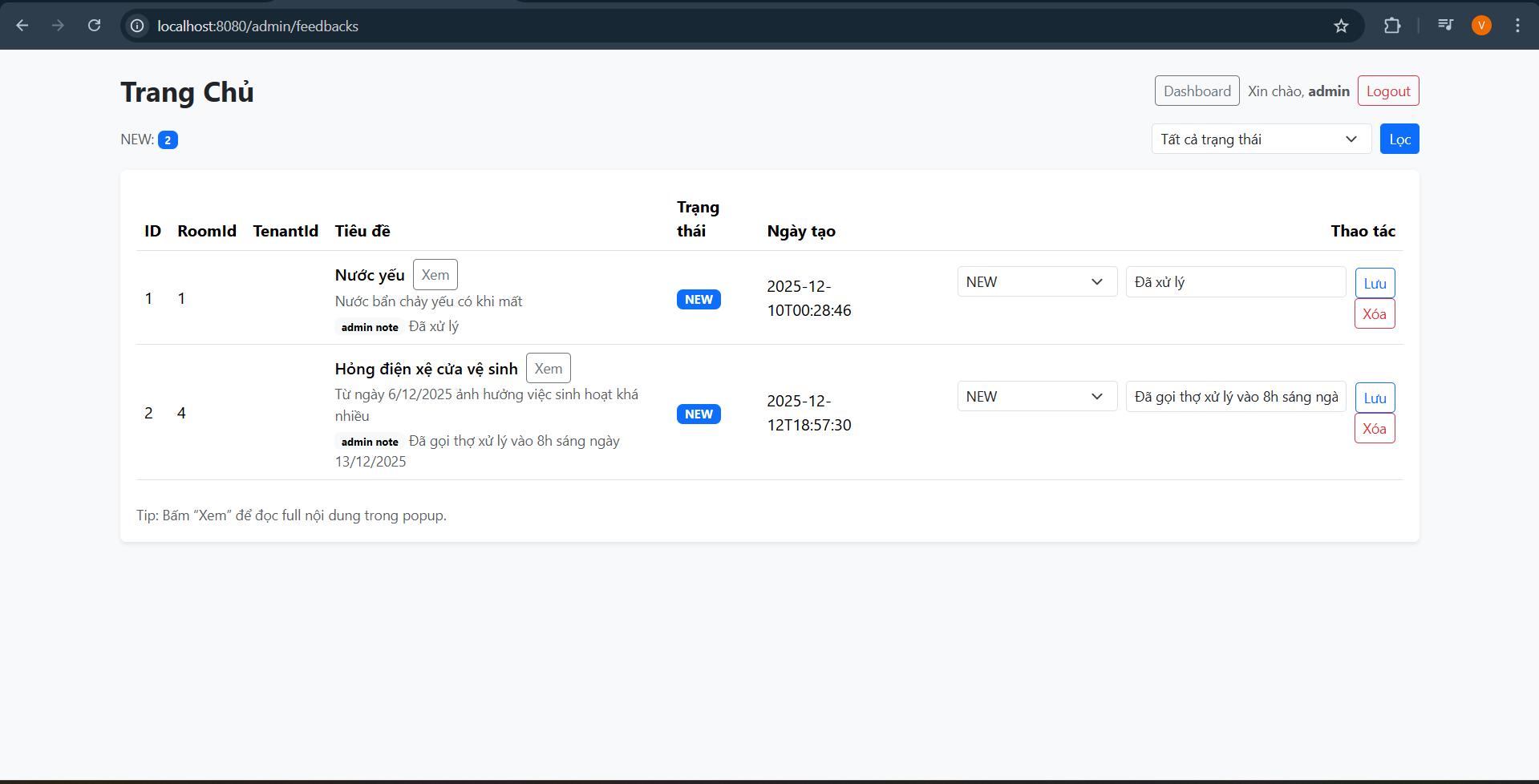
Quản lý chi phí điện nước



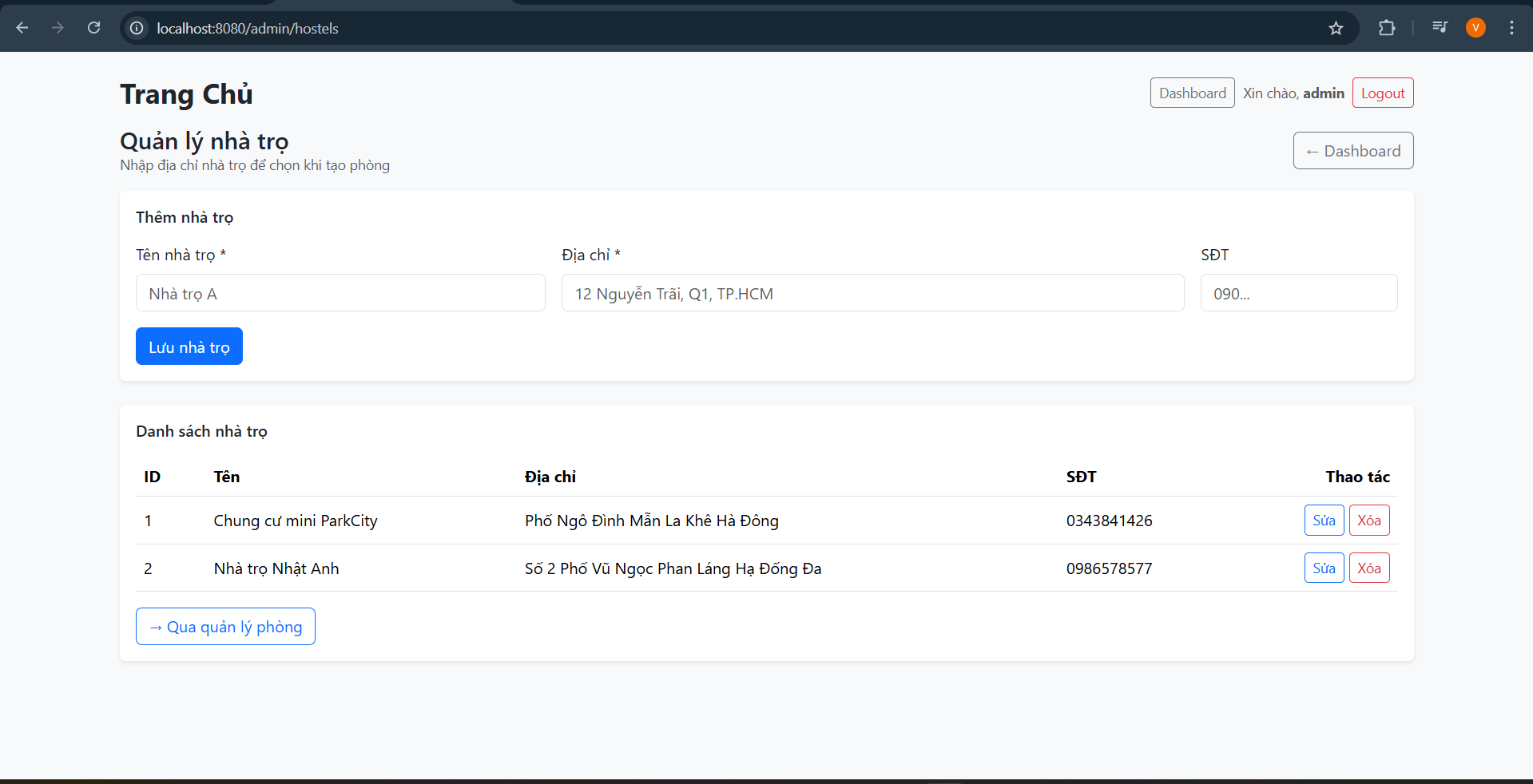
Quản lý hoá đơn



Phản ánh của người dùng



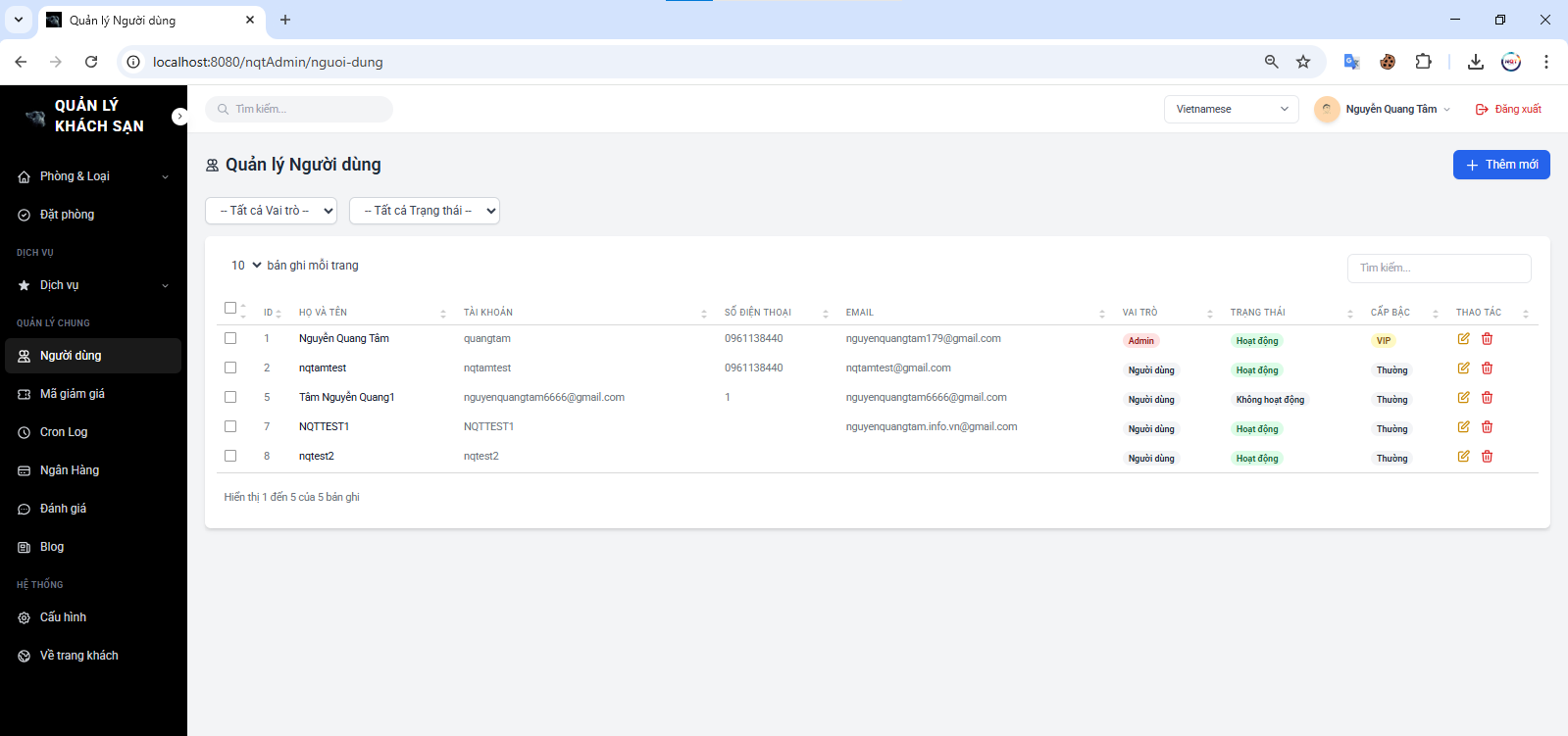
Quản lý nhà trọ



Yêu cầu thanh toán



Danh sách người dùng



# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

Quá trình thực hiện đồ án chuyên đề với đề tài **“Tìm hiểu bài toán quản lý hệ thống phòng trọ”** đã giúp em có cơ hội củng cố và mở rộng các kiến thức đã được học trên giảng đường, đồng thời tiếp cận và vận dụng các kiến thức đó vào một bài toán thực tế cụ thể. Thông qua quá trình nghiên cứu, phân tích và xây dựng hệ thống, em đã rút ra được nhiều kinh nghiệm quý báu, cụ thể như sau:

* + - Nắm được các **quy trình nghiệp vụ cơ bản trong quản lý hệ thống phòng trọ**, bao gồm: quản lý khu nhà trọ, quản lý phòng, quản lý người thuê, lập và quản lý hợp đồng thuê phòng, ghi chỉ số điện nước, lập hóa đơn và quản lý thanh toán.
    - Hiểu rõ hơn về **quy trình phân tích yêu cầu hệ thống**, từ việc khảo sát bài toán thực tế đến việc thiết kế cơ sở dữ liệu phù hợp, đảm bảo tính chặt chẽ và nhất quán của dữ liệu
    - Củng cố và nâng cao kiến thức về **cơ sở dữ liệu**, mô hình hóa dữ liệu, mối quan hệ giữa các bảng, cũng như kỹ năng xây dựng các chức năng cơ bản phục vụ cho công tác quản lý.

**Ưu điểm:**

Hệ thống quản lý phòng trọ đã được xây dựng với **các chức năng cơ bản và cần thiết**, đáp ứng được nhu cầu quản lý thông tin phòng trọ, người thuê, hợp đồng và hóa đơn.

Cơ sở dữ liệu được thiết kế **tương đối đầy đủ, logic**, đảm bảo tính toàn vẹn và hạn chế trùng lặp dữ liệu.

Hệ thống giúp **giảm bớt công việc thủ công**, nâng cao hiệu quả quản lý, đồng thời tạo sự thuận tiện cho người quản lý trong quá trình theo dõi và vận hành.

**Nhược điểm:**

Bên cạnh những kết quả đạt được, hệ thống vẫn còn tồn tại một số hạn chế:

* + - Giao diện hệ thống do thời gian và kinh nghiệm còn hạn chế nên **chưa được hoàn thiện và chưa thật sự thân thiện với người dùng**.
    - Một số chức năng vẫn còn đơn giản, **chưa tối ưu về trải nghiệm sử dụng** và hiệu suất xử lý.
    - Hệ thống chưa tích hợp nhiều tính năng nâng cao để phục vụ nhu cầu quản lý chuyên sâu.

**SHướng phát triển:**

Trong thời gian tới, hệ thống quản lý phòng trọ có thể được tiếp tục hoàn thiện và phát triển theo các hướng sau:

* + - Cải thiện và khắc phục các lỗi phát sinh trong quá trình sử dụng, đảm bảo hệ thống hoạt động ổn định hơn.
    - Nâng cấp giao diện theo hướng **trực quan, thân thiện và dễ sử dụng**.
    - Bổ sung thêm các chức năng mới như: thống kê – báo cáo doanh thu, quản lý bảo trì phòng, thông báo hóa đơn tự động, tích hợp thanh toán trực tuyến.
    - Tối ưu hiệu năng và tăng cường bảo mật dữ liệu nhằm đáp ứng tốt hơn yêu cầu thực tế.

## 5.2 Checklist (bảng kiểm tra chức năng)

### 5.2.1 Kiểm tra dữ liệu nhập

|  |  |  |
| --- | --- | --- |
| **TT** | **Các thành phần** | **Tình trạng** |
| 1 | Các trường trên form không được trống theo yêu cầu | OK |
| 2 | Dữ liệu nhập phù hợp (Chuỗi, số, ngày, …) | OK |
| 3 | Đăng ký, đăng nhập tàì khoản | OK |
| 4 | Các ràng buộc (thêm, sửa, xóa) | OK |
| 5 | Kiểm tra dữ liệu nhập trên form đầy đủ | OK |
| 6 | Xử lý dữ liệu trên giỏ hàng đầy đủ | OK |

### 5.2.2 Kiểm tra các liên kết

|  |  |  |
| --- | --- | --- |
| **TT** | **Các liên kết** | **Tình trạng** |
| 1 | Tất cả các trang đều hoạt động | OK |
| 2 | Tất cả các liên kết đều hoạt động đúng | OK |
| 3 | Các trang đều không có lỗi chính tả | OK |
| 4 | Các mẫu sử dụng trên mỗi trang đều phù hợp | OK |
| 5 | Các thành phần trong trang đều hoạt động | OK |
| 6 | Ứng dụng tương thích với nhiều trình duyệt phổ biến | OK |
| 7 | Dữ liệu sau đều có nhắc nhở, thông báo | OK |
| 8 | Các trang đều có liên kết đến trang khác | OK |

# DANH MỤC TÀI LIỆU THAM KHẢO

* Object Management Group (OMG), *UML 2.x Specification*.  
  <https://www.omg.org/spec/UML>
* Draw.io Documentation.  
  <https://www.drawio.com>
* Bootstrap 5 Documentation.  
  https://getbootstrap.com/docs/5.3
* Google Material Design Guidelines.  
  <https://m3.material.io>
* Spring Boot Reference Documentation.  
  https://docs.spring.io/spring-boot/docs/current/reference/html
* Spring Data JPA Documentation.  
  <https://docs.spring.io/spring-data/jpa/docs/current/reference/html>
* Hibernate ORM Documentation.
* MySQL Workbench – EER Diagram Documentation.  
  <https://dev.mysql.com/doc/workbench/en>
* Spring Security Documentation.  
  <https://docs.spring.io/spring-security/reference>
* Phongtro123.com – <https://phongtro123.com>
* Tromoi.com – <https://tromoi.com>
* Homedy.com – <https://homedy.com>